Nicu Sebe
Michael S. Lew
Thomas S. Huang (

Computer V
in Human-
Interaction

LNCS 3058

ECCV 2004 Workshop on HC
Prague, Czech Republic, Ma
Proceedings

Lecture Notes in Computer Science 3058

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

Springer
Berlin
Heidelberg
New York
Hong Kong
London
Milan
Paris
Tokyo

Nicu Sebe Michael S. Lew
Thomas S. Huang (Eds.)

Computer Vision in Human-Computer Interaction

ECCV 2004 Workshop on HCI
Prague, Czech Republic, May 16, 2004
Proceedings

Springer

Volume Editors

Nicu Sebe
University of Amsterdam, Faculty of Science
Kruislaan 403, 1098 SJ Amsterdam, The Netherlands
E-mail: nicu@science.uva.nl

Michael S. Lew
LIACS Media Lab, Leiden University
Niels Bohrweg 1, 2333 CA Leiden, The Netherlands
E-mail: mlew@liacs.nl

Thomas S. Huang
University of Illinois at Urbana-Champaign, Beckman Institute
405 North Mathews Avenue, Urbana, IL 61801, USA
E-mail: huang@ifp.uiuc.edu

Library of Congress Control Number: 2004105047

CR Subject Classification (1998): I.4, I.5, I.3, H.5.2-3

ISSN 0302-9743
ISBN 3-540-22012-7 Springer-Verlag Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by DA-TeX Gerd Blumenstein
Printed on acid-free paper SPIN: 11008422 06/3142 5 4 3 2 1 0

Preface

Human-Computer Interaction (HCI) lies at the crossroads of many scientific areas including artificial intelligence, computer vision, face recognition, motion tracking, etc. In order for HCI systems to interact seamlessly with people, they need to understand their environment through vision and auditory input. Moreover, HCI systems should learn how to adaptively respond depending on the situation.

The goal of this workshop was to bring together researchers from the field of computer vision whose work is related to human-computer interaction. The articles selected for this workshop address a wide range of theoretical and application issues in human-computer interaction ranging from human-robot interaction, gesture recognition, and body tracking, to facial features analysis and human-computer interaction systems.

This year 45 papers from 18 countries were submitted and 19 were accepted for presentation at the workshop after being reviewed by at least 3 members of the Program Committee.

We would like to thank all members of the Program Committee, as well as the additional reviewers listed below, for their help in ensuring the quality of the papers accepted for publication. We are grateful to Prof. Kevin Warwick for giving the keynote address.

In addition, we wish to thank the organizers of the 8th European Conference on Computer Vision (ECCV 2004) and our sponsors, the University of Amsterdam, the Leiden Institute of Advanced Computer Science, and the University of Illinois at Urbana-Champaign, for support in setting up our workshop.

March 2004

Nicu Sebe
Michael S. Lew
Thomas S. Huang

International Workshop on Human-Computer Interaction 2004 (HCI 2004) Organization

Organizing Committee

Nicu Sebe	University of Amsterdam, The Netherlands
Michael S. Lew	Leiden University, The Netherlands
Thomas S. Huang	University of Illinois at Urbana-Champaign, USA

Program Committee

Kiyo Aizawa	University of Tokyo, Japan
Alberto Del Bimbo	University of Florence, Italy
Tat-Seng Chua	National University of Singapore, Singapore
Roberto Cipolla	University of Cambridge, UK
Ira Cohen	HP Research Labs, USA
James Crowley	INRIA Rhônes Alpes, France
Marc Davis	University of California at Berkeley, USA
Ashutosh Garg	IBM Research, USA
Theo Gevers	University of Amsterdam, The Netherlands
Alan Hanjalic	TU Delft, The Netherlands
Thomas S. Huang	University of Illinois at Urbana-Champaign, USA
Alejandro Jaimes	FujiXerox, Japan
Michael S. Lew	Leiden University, The Netherlands
Jan Nesvadba	Philips Research, The Netherlands
Alex Pentland	Massachusetts Institute of Technology, USA
Rosalind Picard	Massachusetts Institute of Technology, USA
Stan Sclaroff	Boston University, USA
Nicu Sebe	University of Amsterdam, The Netherlands
John R. Smith	IBM Research, USA
Hari Sundaram	Arizona State University, USA
Qi Tian	University of Texas at San Antonio, USA
Guangyou Xu	Tsinghua University, China
Ming-Hsuan Yang	Honda Research Labs, USA
HongJiang Zhang	Microsoft Research Asia, China
Xiang (Sean) Zhou	Siemens Research, USA

Additional Reviewers

Preetha Appan	Arizona State University
Marco Bertini	University of Florence
Yinpeng Chen	Arizona State University
Yunqiang Chen	Siemens Research
Vidyarani Dyaberi	Arizona State University
Murat Erdem	Boston University
Ashish Kapoor	Massachusetts Institute of Technology
Shreeharsh Kelkar	Arizona State University
Rui Li	Boston University
Zhu Li	Northwestern University
Ankur Mani	Arizona State University
Yelizaveta Marchenko	National University of Singapore
Teck-Khim Ng	National University of Singapore
Tat Hieu Nguyen	University of Amsterdam
Walter Nunziati	University of Florence
Maja Pantic	TU Delft
Bageshree Shevade	Arizona State University
Harini Sridharan	Arizona State University
Taipeng Tian	Boston University
Alessandro Valli	University of Florence
Lei Wang	Tsinghua University
Joost van de Weijer	University of Amsterdam
Bo Yang	Tsinghua University
Yunlong Zhao	National University of Singapore
Hanning Zhou	University of Illinois at Urbana-Champaign

Sponsors

Faculty of Science, University of Amsterdam
The Leiden Institute of Advanced Computer Science, Leiden University
Beckman Institute, University of Illinois at Urbana-Champaign

Table of Contents

Systems

Face and Head

The State-of-the-Art in Human-Computer Interaction

Nicu Sebe[1], Michael S. Lew[2], and Thomas S. Huang[3]

[1] Faculty of Science, University of Amsterdam, The Netherlands
[2] LIACS Media Lab, Leiden University, The Netherlands
[3] Beckman Institute, University of Illinois at Urbana-Champaign, USA

Human computer interaction (HCI) lies at the crossroads of many scientific areas including artificial intelligence, computer vision, face recognition, motion tracking, etc. In recent years there has been a growing interest in improving all aspects of the interaction between humans and computers. It is argued that to truly achieve effective human-computer intelligent interaction (HCII), there is a need for the computer to be able to interact naturally with the user, similar to the way human-human interaction takes place.

Humans interact with each other mainly through speech, but also through body gestures, to emphasize a certain part of the speech and display of emotions. As a consequence, the new interface technologies are steadily driving toward accommodating information exchanges via the natural sensory modes of sight, sound, and touch. In face-to-face exchange, humans employ these communication paths simultaneously and in combination, using one to complement and enhance another. The exchanged information is largely encapsulated in this natural, multimodal format. Typically, conversational interaction bears a central burden in human communication, with vision, gaze, expression, and manual gesture often contributing critically, as well as frequently embellishing attributes such as emotion, mood, attitude, and attentiveness. But the roles of multiple modalities and their interplay remain to be quantified and scientifically understood. What is needed is a science of human-computer communication that establishes a framework for multimodal "language" and "dialog", much like the framework we have evolved for spoken exchange.

Another important aspect is the development of Human-Centered Information Systems. The most important issue here is how to achieve synergism between man and machine. The term "Human-Centered" is used to emphasize the fact that although all existing information systems were designed with human users in mind, many of them are far from being user friendly. What can the scientific/engineering community do to effect a change for the better?

Information systems are ubiquitous in all human endeavors including scientific, medical, military, transportation, and consumer. Individual users use them for learning, searching for information (including data mining), doing research (including visual computing), and authoring. Multiple users (groups of users, and groups of groups of users) use them for communication and collaboration. And either single or multiple users use them for entertainment. An information system consists of two components: Computer (data/knowledge base, and information processing engine), and humans. It is the intelligent interaction between

N. Sebe et al. (Eds.): HCI/ECCV 2004, LNCS 3058, pp. 1–6, 2004.

the two that we are addressing. We aim to identify the important research issues, and to ascertain potentially fruitful future research directions. Furthermore, we shall discuss how an environment can be created which is conducive to carrying out such research.

In many important HCI applications such as computer aided tutoring and learning, it is highly desirable (even mandatory) that the response of the computer take into account the emotional or cognitive state of the human user. Emotions are displayed by visual, vocal, and other physiological means. There is a growing amount of evidence showing that emotional skills are part of what is called "intelligence" [1, 2]. Computers today can recognize much of what is said, and to some extent, who said it. But, they are almost completely in the dark when it comes to how things are said, the affective channel of information. This is true not only in speech, but also in visual communications despite the fact that facial expressions, posture, and gesture communicate some of the most critical information: how people feel. Affective communication explicitly considers how emotions can be recognized and expressed during human-computer interaction.

In most cases today, if you take a human-human interaction, and replace one of the humans with a computer, then the affective communication vanishes. Furthermore, it is not because people stop communicating affect - certainly we have all seen a person expressing anger at his machine. The problem arises because the computer has no ability to recognize if the human is pleased, annoyed, interested, or bored. Note that if a human ignored this information, and continued babbling long after we had yawned, we would not consider that person very intelligent. Recognition of emotion is a key component of intelligence. Computers are presently affect-impaired.

Furthermore, if you insert a computer (as a channel of communication) between two or more humans, then the affective bandwidth may be greatly reduced. Email may be the most frequently used means of electronic communication, but typically all of the emotional information is lost when our thoughts are converted to the digital media.

Research is therefore needed for new ways to communicate affect through computer-mediated environments. Computer-mediated communication today almost always has less affective bandwidth than "being there, face-to-face". The advent of affective wearable computers, which could help amplify affective information as perceived from a person's physiological state, are but one possibility for changing the nature of communication.

The papers in the proceedings present specific aspects of the technologies that support human-computer interaction. Most of the authors are computer vision researchers whose work is related to human-computer interaction.

The paper by Warwick and Gasson [3] discusses the efficacy of a direct connection between the human nervous system and a computer network. The authors give an overview of the present state of neural implants and discuss the possibilities regarding such implant technology as a general purpose human-computer interface for the future.

Human-robot interaction (HRI) has recently drawn increased attention. Autonomous mobile robots can recognize and track a user, understand his verbal commands, and take actions to serve him. A major reason that makes HRI distinctive from traditional HCI is that robots can not only passively receive information from environment but also make decisions and actively change the environment. An interesting approach in this direction is presented by Huang and Weng [4]. Their paper presents a motivational system for HRI which integrates novelty and reinforcement learning. The robot develops its motivational system through its interactions with the world and the trainers. A vision-based gestural guidance interface for mobile robotic platforms is presented by Paquin and Cohen [5]. The interface controls the motion of the robot by using a set of predefined static and dynamic hand gestures inspired by the marshaling code. Images captured by an on-board camera are processed in order to track the operator's hand and head. A similar approach is taken by Nickel and Stiefelhagen [6]. Given the images provided by a calibrated stereo-camera, color and disparity information are integrated into a multi-hypotheses tracking framework in order to find the 3D positions of the respective body parts. Based on the motion of the hands, an HMM-based approach is applied to recognize pointing gestures.

Mixed reality (MR) opens a new direction for human-computer interaction. Combined with computer vision techniques, it is possible to create advanced input devices. Such a device is presented by Tosas and Li [7]. They describe a virtual keypad application which illustrates the virtual touch screen interface idea. Visual tracking and interpretation of the user's hand and finger motion allows the detection of key presses on the virtual touch screen. An interface tailored to create a design-oriented realistic MR workspace is presented by Gheorghe, et al. [8]. An augmented reality human computer interface for object localization is presented by Siegl, et al. [9]. A 3D pointing interface that can perform 3D recognition of arm pointing direction is proposed by Hosoya, et al. [10]. A hand gesture recognition system is also proposed by Licsár and Szirányi [11]. A hand pose estimation approach is discussed by Stenger, et al. [12]. They present an analysis of the design of classifiers for use in a more general hierarchical object recognition approach.

The current down-sizing of computers and sensory devices allows humans to wear these devices in a manner similar to clothes. One major direction of wearable computing research is to smartly assist humans in daily life. Yamazoe, et al. [13] propose a body attached system to capture audio and visual information corresponding to user experience. This data contains significant information for recording/analyzing human activities and can be used in a wide range of applications such as digital diary or interaction analysis. Another wearable system is presented by Tsukizawa, et al. [14].

3D head tracking in a video sequence has been recognized as an essential prerequisite for robust facial expression/emotion analysis, face recognition and model-based coding. The paper by Dornaika and Ahlberg [15] presents a system for real-time tracking of head and facial motion using 3D deformable models. A similar system is presented by Sun, et al [16]. Their goal is to use their real-

time tracking system to recognize authentic facial expressions. A pose invariant face recognition approach is proposed by Lee and kim [17]. A 3D head pose estimation approach is proposed by Wang, et al [18]. They present a new method for computing the head pose by using projective invariance of the vanishing point. A multi-view face image synthesis using a factorization model is introduced by Du and Lin [19]. The proposed method can be applied to a several HCI areas such as view independent face recognition or face animation in a virtual environment.

The emerging idea of ambient intelligence is a new trend in human-computer interaction. An ambient intelligence environment is sensitive to the presence of people and responsive to their needs. The environment will be capable of greeting us when we get home, of judging our mood and adjusting our environment to reflect it. Such an environment is still a vision but it is one that struck a chord in the minds of researchers around the world and become the subject of several major industry initiatives. One such initiative is presented by Kleindienst, et al. [20]. They use speech recognition and computer vision to model new generation of interfaces in the residential environment. An important part of such a system is the localization module. A possible implementation of this module is proposed by Okatani and Takuichi [21]. Another important part of an ambient intelligent system is the extraction of typical actions performed by the user. A solution to this problem is provided by Ma and Lin [22].

Human-computer interaction is a particularly wide area which involves elements from diverse areas such as psychology, ergonomics, engineering, artificial intelligence, databases, etc. This proceedings represents a snapshot of the state of the art in human computer interaction with an emphasis on intelligent interaction via computer vision, artificial intelligence, and pattern recognition methodology. Our hope is that in the not too distant future the research community will have made significant strides in the science of human-computer interaction, and that new paradigms will emerge which will result in natural interaction between humans, computers, and the environment.

References

[1] Salovey, P., Mayer, J.: Emotional intelligence. Imagination, Cognition, and Personality **9** (1990) 185–211 2
[2] Goleman, D.: Emotional Intelligence. Bantam Books, New York (1995) 2
[3] Warwick, K., Gasson, M.: Practical interface experiments with implant technology. In: International Workshop on Human-Computer Interaction, Lecture Notes in Computer Science, vol. 3058, Springer (2004) 6–16 2
[4] Huang, X., Weng, J.: Motivational system for human-robot interaction. In: International Workshop on Human-Computer Interaction, Lecture Notes in Computer Science, vol. 3058, Springer (2004) 17–27 3
[5] Paquin, V., Cohen, P.: A vision-based gestural guidance interface for mobile robotic platforms. In: International Workshop on Human-Computer Interaction, Lecture Notes in Computer Science, vol. 3058, Springer (2004) 38–46 3
[6] Nickel, K., Stiefelhagen, R.: Real-time person tracking and pointing gesture recognition for human-robot interaction. In: International Workshop on Human-Computer Interaction, Lecture Notes in Computer Science, vol. 3058, Springer (2004) 28–37 3

[7] Tosas, M., Li, B.: Virtual touch screen for mixed reality. In: International Workshop on Human-Computer Interaction, Lecture Notes in Computer Science, vol. 3058, Springer (2004) 47–57 3
[8] Gheorghe, L., Ban, Y., Uehara, K.: Exploring interactions specific to mixed reality 3D modeling systems. In: International Workshop on Human-Computer Interaction, Lecture Notes in Computer Science, vol. 3058, Springer (2004) 113–123 3
[9] Siegl, H., Schweighofer, G., Pinz, A.: An AR human computer interface for object localization in a cognitive vision framework. In: International Workshop on Human-Computer Interaction, Lecture Notes in Computer Science, vol. 3058, Springer (2004) 167–177 3
[10] Hosoya, E., Sato, H., Kitabata, M., Harada, I., Nojima, H., Onozawa, A.: Arm-pointer: 3D pointing interface for real-world interaction. In: International Workshop on Human-Computer Interaction, Lecture Notes in Computer Science, vol. 3058, Springer (2004) 70–80 3
[11] Licsár, A., Szirányi, T.: Hand gesture recognition in camera-projector system. In: International Workshop on Human-Computer Interaction, Lecture Notes in Computer Science, vol. 3058, Springer (2004) 81–91 3
[12] Stenger, B., Thayananthan, A., Torr, P., Cipolla, R.: Hand pose estimation using hierarchical detection. In: International Workshop on Human-Computer Interaction, Lecture Notes in Computer Science, vol. 3058, Springer (2004) 102–112 3
[13] Yamazoe, H., Utsumi, A., Tetsutani, N., Yachida, M.: A novel wearable system for capturing user view images. In: International Workshop on Human-Computer Interaction, Lecture Notes in Computer Science, vol. 3058, Springer (2004) 156–166 3
[14] Tsukizawa, S., Sumi, K., Matsuyama, T.: 3D digitization of a hand-held object with a wearable vision sensor. In: International Workshop on Human-Computer Interaction, Lecture Notes in Computer Science, vol. 3058, Springer (2004) 124–134 3
[15] Dornaika, F., Ahlberg, J.: Model-based head and facial motion tracking. In: International Workshop on Human-Computer Interaction, Lecture Notes in Computer Science, vol. 3058, Springer (2004) 211–221 3
[16] Sun, Y., Sebe, N., Lew, M., Gevers, T.: Authentic emotion detection in real-time video. In: International Workshop on Human-Computer Interaction, Lecture Notes in Computer Science, vol. 3058, Springer (2004) 92–101 3
[17] Lee, H. S., Kim, D.: Pose invariant face recognition using linear pose transformation in feature space. In: International Workshop on Human-Computer Interaction, Lecture Notes in Computer Science, vol. 3058, Springer (2004) 200–210 4
[18] Wang, J. G., Sung, E., Venkateswarlu, R.: EM enhancement of 3D head pose estimated by perspective invariance. In: International Workshop on Human-Computer Interaction, Lecture Notes in Computer Science, vol. 3058, Springer (2004) 178–188 4
[19] Du, Y., Lin, X.: Multi-view face image synthesis using factorization model. In: International Workshop on Human-Computer Interaction, Lecture Notes in Computer Science, vol. 3058, Springer (2004) 189–199 4
[20] Kleindienst, J., Macek, T., Serédi, L., Šedivý, J.: Djinn: Interaction framework for home environment using speech and vision. In: International Workshop on Human-Computer Interaction, Lecture Notes in Computer Science, vol. 3058, Springer (2004) 145–155 4

[21] Okatani, I., Takuichi, N.: Location-based information support system using multiple cameras and LED light sources with the compact battery-less information terminal (CoBIT). In: International Workshop on Human-Computer Interaction, Lecture Notes in Computer Science, vol. 3058, Springer (2004) 135–144 4

[22] Ma, G., Lin, X.: Typical sequences extraction and recognition. In: International Workshop on Human-Computer Interaction, Lecture Notes in Computer Science, vol. 3058, Springer (2004) 58–69 4

Practical Interface Experiments with Implant Technology

Kevin Warwick and Mark Gasson

Department of Cybernetics, University of Reading
Whiteknights, Reading, RG6 6AY, UK
{k.warwick,m.n.gasson}@reading.ac.uk

Abstract. In this paper results are shown to indicate the efficacy of a direct connection between the human nervous system and a computer network. Experimental results obtained thus far from a study lasting for over 3 months are presented, with particular emphasis placed on the direct interaction between the human nervous system and a piece of wearable technology. An overview of the present state of neural implants is given, as well as a range of application areas considered thus far. A view is also taken as to what may be possible with implant technology as a general purpose human-computer interface for the future.

1 Introduction

Biological signals can be recorded in a number of ways and can then be acted upon in order to control or manipulate an item of technology, or purely for monitoring purposes, e.g. [1, 2]. However, in the vast majority of cases, these signals are collected externally to the body and, whilst this is positive from the viewpoint of non-intrusion into the body with potential medical side-effects, it does present enormous problems in deciphering and understanding the signals obtained [3, 4]. In particular, noise issues can override all other, especially when collective signals are all that can be recorded, as is invariably the case with neural recordings. The main issue is selecting exactly which signals contain useful information and which are noise. In addition, if stimulation of the nervous system is required, this, to all intents and purposes, is not possible in a meaningful way with external connections. This is mainly due to the strength of signal required, making stimulation of unique or even small subpopulations of sensory receptor or motor unit channels unachievable by such a method.

1.1 Background

A number of researchers have concentrated on animal (non-human) studies which have certainly provided results that contribute to the knowledge base of the field. Human studies however are unfortunately relatively limited in number, although it could be said that research into wearable computers has provided some evidence of what can be done technically with bio-signals. Whilst augmenting shoes and glasses with microcomputers [5] are perhaps not directly useful for our studies, monitoring indications of stress and alertness can be helpful, with the state of the wearable device altered to affect the wearer. Also of relevance here are studies in which a miniature computer screen was fitted onto a standard pair of glasses. In this research the wearer

N. Sebe et al. (Eds.): HCI/ECCV 2004, LNCS 3058, pp. 7-16, 2004.

was given a form of augmented/remote vision [6], where information about a remote scene could be relayed back to the wearer. However, wearable computers require some form of signal conversion to take place in order to interface the external technology with the specific human sensory receptors. Of much more interest to our own studies are investigations in which a direct electrical link is formed between the nervous system and technology.

Numerous relevant animal studies have been carried out, see [7] for a review. For example, in one reported study the extracted brain of a lamprey was used to control the movement of a small-wheeled robot to which it was attached [8]. The innate response of a lamprey is to position itself in water by detecting and reacting to external light on the surface of the water. The lamprey robot was surrounded by a ring of lights and the innate behaviour was employed to cause the robot to move swiftly around towards the appropriate light source, when different lights were switched on and off.

Several studies have involved rats as the subjects. In one of these [9], rats were taught to pull a lever such that they received a liquid treat as a reward for their efforts. Electrodes were chronically implanted into the motor cortex of the rats' brains to directly detect neural signals generated when each rat (it is claimed) thought about pulling the lever, but, importantly, before any physical movement occurred. These signals were used to directly release the reward before a rat actually carried out the physical action of pulling the lever. Over the time of the trial, which lasted for a few days, four of the six implanted rats learned that they need not actually initiate any action in order to obtain the reward; merely thinking about the action was sufficient. One point of note here is that although the research is certainly of value, because rats were employed in the trial we cannot be sure what they were actually thinking in order to receive the reward.

Meanwhile, in another study [10], the brains of a number of rats were stimulated via electrodes in order to teach them to solve a maze problem. Reinforcement learning was used in the sense that, as it is claimed, pleasurable stimuli were evoked when a rat moved in the correct direction. Again however, we cannot be sure of the actual feelings perceived by the rats, whether they were at all pleasurable when successful or unpleasant when a negative route was taken.

1.2 Human Integration

Studies looking at, in some sense, integrating technology with the Human Central Nervous System range from those which can be considered to be diagnostic [11], to those which are aimed at the amelioration of symptoms [12, 13, 14] to those which are clearly directed towards the augmentation of senses [15, 16]. However, by far the most widely reported research with human subjects is that involving the development of an artificial retina [17]. Here small arrays have been attached to a functioning optic nerve, but where the person concerned has no operational vision. By means of direct stimulation of the nerve with appropriate signal sequences the user has been able to perceive simple shapes and letters. Although relatively successful thus far, this research would appear to have a long way to go.

Electronic neural stimulation has proved to be extremely successful in other areas which can be loosely termed as being restorative. In this class, applications range from cochlea implants to the treatment of Parkinson's disease symptoms. The most relevant to our study here however is the use of a single electrode brain implant, enabling a brainstem stroke victim to control the movement of a cursor on a computer screen [18]. In the first instance extensive functional magnetic resonance imaging (fMRI) of the subject's brain was carried out. The subject was asked to think about moving his hand and the fMRI scanner was used to determine where neural activity was most pronounced. A hollow glass electrode cone containing two gold wires was subsequently positioned into the motor cortex, centrally located in the area of maximum-recorded activity. When the patient thought about moving his hand, the output from the electrode was amplified and transmitted by a radio link to a computer where the signals were translated into control signals to bring about movement of the cursor. The subject learnt to move the cursor around by thinking about different hand movements. No signs of rejection of the implant were observed whilst it was in position [18].

In all of the human studies described, the main aim is to use technology to achieve some restorative functions where a physical problem of some kind exists, even if this results in an alternative ability being generated. Although such an end result is certainly of interest, one of the main directions of the study reported in this paper is to investigate the possibility of giving a human extra capabilities, over and above those initially in place.

In the section which follows a MicroElectrode Array (MEA) of the spiked electrode type is described. An array of this type was implanted into a human nervous system to act as an electrical silicon/biological interface between the human nervous system and a computer. As an example, a pilot study is described in which the output signals from the array are used to drive a wearable computing device in a switching mode. This is introduced merely as an indication of what is possible. It is worth emphasising here that what is described in this article is an actual application study rather than a computer simulation or mere speculation.

2 Invasive Neural Interface

When a direct connection to the human nervous system is required, there are, in general, two approaches for peripheral nerve interfaces: Extraneural and Intraneural. The cuff electrode is the most common extraneural device. By fitting tightly around the nerve trunk, it is possible to record the sum of the single fibre action potentials, known as the compound action potential (CAP). It can also be used for crudely selective neural stimulation of a large region of the nerve trunk. In some cases the cuff can contain a second or more electrodes, thereby allowing for an approximate measurement of signal speed travelling along the nerve fibres.

However, for applications which require a much finer granularity for both selective monitoring and stimulation, an intraneural interface such as single electrodes either individually or in groups can be employed. To open up even more possibilities a MicroElectrode Array (MEA) is well suited. MEAs can take on a number of forms, for example they can be etched arrays that lie flat against a neural surface [19] or

spiked arrays with electrode tips. The MEA employed in this study is of this latter type and contains a total of 100 electrodes which, when implanted, become distributed within the nerve fascicle. In this way, it is possible to gain direct access to nerve fibres from muscle spindles, motor neural signals to particular motor units or sensory receptors. Essentially, such a device allows a bi-directional link between the human nervous system and a computer [20, 21, 22].

2.1 Surgical Procedure

On 14 March 2002, during a 2 hour procedure at the Radcliffe Infirmary, Oxford, a MEA was surgically implanted into the median nerve fibres of the left arm of the first named author (KW). The array measured 4mm x 4mm with each of the electrodes being 1.5mm in length. Each electrode was individually wired via a 20cm wire bundle to an electrical connector pad. A distal skin incision marked at the distal wrist crease medial to the *palmaris longus* tendon was extended approximately 4 cm into the forearm. Dissection was performed to identify the median nerve. In order that the risk of infection in close proximity to the nerve was reduced, the wire bundle was run subcutaneously for 16 cm before exiting percutaneously. As such a second proximal skin incision was made distal to the elbow 4 cm into the forearm. A modified plastic shunt passer was inserted subcutaneously between the two incisions by means of a tunnelling procedure. The MEA was introduced to the more proximal incision and pushed distally along the passer to the distal skin incision such that the wire bundle connected to the MEA ran within it. By removing the passer, the MEA remained adjacent to the exposed median nerve at the point of the first incision with the wire bundle running subcutaneously, exiting at the second incision. At the exit point, the wire bundle linked to the electrical connector pad which remained external to the arm.

The perineurium of the median nerve was dissected under microscope to facilitate the insertion of electrodes and ensure adequate electrode penetration depth. Following dissection of the perineurium, a pneumatic high velocity impact inserter was positioned such that the MEA was under a light pressure to help align insertion direction. The MEA was pneumatically inserted into the radial side of the median nerve allowing the MEA to sit adjacent to the nerve fibres with the electrodes penetrating into a fascicle. The median nerve fascicle selected was estimated to be approximately 4 mm in diameter. Penetration was confirmed under microscope. Two Pt/Ir reference wires were positioned in the fluids surrounding the nerve.

The arrangements described remained permanently in place for 96 days, until 18th June 2002, at which time the implant was removed.

2.2 Neural Stimulation and Neural Recordings

The array, once in position, acted as a bi-directional neural interface. Signals could be transmitted directly from a computer, by means of either a hard wire connection or through a radio transmitter/receiver unit, to the array and thence to directly bring about a stimulation of the nervous system. In addition, signals from neural activity could be detected by the electrodes and sent to the computer. During experimentation, it was found that typical activity on the median nerve fibres occurs around a centroid

frequency of approximately 1 KHz with signals of apparent interest occurring well below 3.5 KHz. However noise is a distinct problem due to inductive pickup on the wires, so had to be severely reduced. To this end a fifth order band limited Butterworth filter was used with corner frequencies of f_{low}= 250 Hz and f_{high} = 7.5 KHz.

To allow freedom of movement, a small wearable signal processing unit with RF communications was developed to be worn on a gauntlet around the wrist. This custom hardware consisted of a 20 way multiplexer, two independent filters, two 10bit A/D converters, a microcontroller and an FM radio transceiver module. Either 1 or 2 electrodes from the array could be quasi-statically selected, digitised and sent over the radio link to a corresponding receiver connected to a PC. At this point they could either be recorded or transmitted further in order to operate networked technology, as described in the following section. Onward transmission of the signal was via an encrypted TCP/IP tunnel, over the local area network, or wider internet. Remote configuration of various parameters on the wearable device was also possible via the radio link from the local PC or the remote PC via the encrypted tunnel.

Stimulation of the nervous system by means of the array was especially problematic due to the limited nature of existing results using this type of interface. Published work is restricted largely to a respectably thorough but short term study into the stimulation of the sciatic nerve in cats [20]. Much experimental time was therefore required, on a trial and error basis, to ascertain what voltage/current relationships would produce a reasonable (i.e. perceivable but not painful) level of nerve stimulation.

Further factors which may well emerge to be relevant, but were not possible to predict in this experimental session were:

(a) The plastic, adaptable nature of the human nervous system, especially the brain – even over relatively short periods.

(b) The effects of movement of the array in relation to the nerve fibres, hence the connection and associated input impedance of the nervous system was not completely stable.

After extensive experimentation it was found that injecting currents below 80μA onto the median nerve fibres had little perceivable effect. Between 80μA and 100μA all the functional electrodes were able to produce a recognisable stimulation, with an applied voltage of around 20 volts peak to peak, dependant on the series electrode impedance. Increasing the current above 100μA had little additional effect; the stimulation switching mechanisms in the median nerve fascicle exhibited a non-linear thresholding characteristic.

In all successful trials, the current was applied as a bi-phasic signal with pulse duration of 200μsec and an inter-phase delay of 100μsec. A typical stimulation waveform of constant current being applied to one of the MEAs implanted electrodes is shown in Fig. 1.

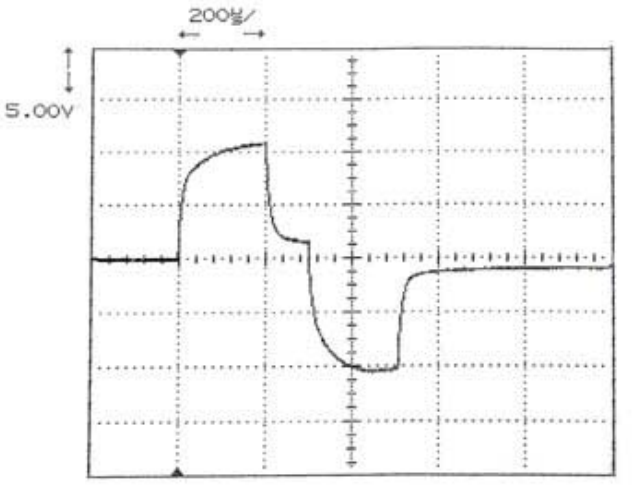


Fig. 1. Voltage profile during one bi-phasic stimulation pulse cycle with a constant current of 80μA

It was therefore possible to create alternative sensations via this new input route to the nervous system, thereby by-passing the normal sensory inputs. It should be noted that it took around 6 weeks for the recipient to recognise the stimulating signals reliably. This time period can be due to a number of contributing factors:

(a) Suitable pulse characteristics, (i.e. amplitude, frequency etc) required to bring about a perceivable stimulation were determined experimentally during this time.
(b) The recipient's brain had to adapt to recognise the new signals it was receiving.
(c) The bond between the recipient's nervous system and the implant was physically changing.

3 Neural Interaction with Wearable Technology

An experiment was conducted to utilise neural signals directly to control the visual effect produced by a specially constructed necklace. The necklace (Fig. 2.) was conceptualised by the Royal College of Art, London, and constructed in the Department of Cybernetics in Reading University. The main visual effect of the jewellery was the use of red and blue light emitting diodes (LEDs) interspersed within the necklace frame such that the main body of the jewellery could appear red, blue or by amplitude modulation of the two colours, a range of shades between the two.

Fig. 2. Wearable Jewellery interacting with the human nervous system

Neural signals taken directly from the recipient's nervous system were employed to operate the LEDs within the necklace in real-time. With fingers operated such that the hand was completely clasped, the LEDs shone bright red, while with fingers opened, as in Fig. 2., the LEDs shone bright blue. The jewellery could either be operated so that the LEDs merely switched between extremes of red and blue or conversely intermediate shades of purple would be seen to indicate the degree of neural activity. Reliability of operation was however significantly higher with the first of these scenarios, possibly due to the use of nonlinear thresholding to cause jewellery action.

4 Application Range

One application of the implant has been described in the previous section in order to link this work more directly with ongoing wearable computing research, such as that described in the Introduction to this paper. It is however apparent that the neural signals obtained through the implant can be used for a wide variety of purposes. One of the key aims of this research was, in fact, to assess the feasibility of the implant for use with individuals who have limited functions due to a spinal injury. Hence in other experimental tests, neural signals were employed to control the functioning of a robotic hand and to drive a wheelchair around successfully [20, 22]. The robotic hand was also controlled, via the internet, at a remote location [23].

Once stimulation of the nervous system had been achieved, as described in section 2, the bi-directional nature of the implant could be more fully experimented with. Stimulation of the nervous system was activated by taking signals from fingertips sensors on the robotic hand. So as the robotic hand gripped an object, in response to outgoing neural signals via the implant, signals from the fingertips of the robotic hand brought about stimulation. As the robotic hand applied more pressure the frequency of stimulation increased [23]. The robotic hand was, in this experiment, acting as a remote, extra hand.

In another experiment, signals were obtained from ultrasonic sensors fitted to a baseball cap. The output from these sensors directly affected the rate of neural stimulation. With a blindfold on, the recipient was able to walk around in a cluttered environment whilst detecting objects in the vicinity through the (extra) ultrasonic sense. With no objects nearby, no neural stimulation occurred. As an object moved relatively closer, so the stimulation increased proportionally [24].

It is clear that just about any technology, which can be networked in some way, can be switched on and off and ultimately controlled directly by means of neural signals through an interface such as the implant used in this experimentation. Not only that, but because a bi-directional link has been formed, feedback directly to the brain can increase the range of sensory capabilities. Potential application areas are therefore considerable.

5 Discussion

This study was partly carried out to assess the usefulness of an implanted interface to help those with a spinal injury. It can be reported that there was, during the course of the study, no sign of infection and the recipient's body, far from rejecting the implant, appeared to accept the implant fully. Indeed, results from the stimulation study indicate that acceptance of the implant could well have been improving over time.

Certainly such an implant would appear to allow for, in the case of those with a spinal injury, the restoration of some, otherwise missing, movement; the return of the control of body functions to the body's owner; or for the recipient to control technology around them. This, however, will have to be further established through future human trials.

But such implanted interface technology would appear to open up many more opportunities. In the case of the experiments described, an articulated robot hand was controlled directly by neural signals. For someone who has had their original hand amputated this opens up the possibility of them ultimately controlling an articulated hand, as though it were their own, by the power of their own thought.

In terms of the specific wearable application described and pictured in this paper, direct nervous system connections open up a plethora of possibilities. If body state information can be obtained relatively easily, then information can be given, externally of the present condition of an individual. This could be particularly useful for those in intensive care. Emotional signals, in the sense of physical indications of emotions, would also appear to be a possible source of decision switching for external wearables. Not only stress and anger, but also excitement and arousal would appear to be potential signals.

As far as wearables are concerned, this study throws up an important question in terms of who exactly is doing the wearing. By means of a radio link, neural signals from one person can be transmitted remotely to control a wearable on another individual. Indeed this was the experiment successfully carried out and described in this paper. In such cases the wearable is giving indicative information externally, but it may well not be information directly relating to the actual wearer, rather it may be information for the wearer from a remote source.

The authors accept the fact that this is a one off study based on only one implant recipient. It may be that other recipients react in other ways and the experiments carried out would not be so successful with an alternative recipient. In that sense the authors wish this study to be seen as evidence that the concept does work well. Further human trials will be necessary to investigate the breadth of usefulness.

It is recognized that, as far as an implant interface is concerned, what has been achieved is a primitive first step. Indeed, it may well be the case that implants of the type used here are not ultimately those selected for a good link between a computer and the human brain. Nevertheless the results obtained are extremely encouraging.

Acknowledgements

Ethical approval for this research to proceed was obtained from the Ethics and Research Committee at the University of Reading and, in particular with regard to the neurosurgery, by the Oxfordshire National Health Trust Board overseeing the Radcliffe Infirmary, Oxford, UK.

Our thanks go to Mr. Peter Teddy and Mr. Amjad Shad who performed the neurosurgery at the Radcliffe Infirmary and ensured the medical success of the project. Our gratitude is also extended to NSIC, Stoke Mandeville and to the David Tolkien Trust for their support.

We also wish to extend our gratitude to Sompit Fusakul of the Royal College of Art, London who added artistic design to the jewellery employed for the wearable computing experiment.

References

[1] Penny, W., Roberts, S., Curran, E., and Stokes, M., "EEG-based communication: A pattern recognition approach", IEEE Transactions on Rehabilitation Engineering., Vol. 8, Issue.2, pp. 214-215, 2000.

[2] Roberts, S., Penny, W., and Rezek, I., "Temporal and spatial complexity measures for electroencephalogram based brain-computer interfacing", Medical and Biological Engineering and Computing, Vol.37, Issue.1, pp.93-98, 1999.

[3] Wolpaw, J., McFarland, D., Neat, G. and Forneris, C., "An EEG based brain-computer interface for cursor control", Electroencephalography and Clinical Neurophysiology, Vol. 78, Issue.3, pp. 252-259, 1991.

[4] Kubler, A., Kotchoubey, B., Hinterberger, T., Ghanayim, N., Perelmouter, J., Schauer, M., Fritsch, C., Taub, E. and Birbaumer, N., "The Thought Translation device: a neurophysiological approach to communication in total motor paralysis", Experimental Brain Research, Vol. 124, Issue.2, pp. 223-232, 1999.

[5] Thorp, E., "The invention of the first wearable computer", In Proceedings of the Second IEEE International Symposium on Wearable Computers, pp.4–8, Pittsburgh, October 1998.

[6] Mann, S., "Wearable Computing: A first step towards personal imaging", Computer, Vol. 30, Issue.2, pp. 25-32, 1997.

[7] Warwick, K., "I, Cyborg", University of Illinois Press, 2004.

[8] Reger, B., Fleming, K., Sanguineti, V., Simon Alford, S., Mussa-Ivaldi, F., "Connecting Brains to Robots: The Development of a Hybrid System for the Study of Learning in Neural Tissues", Artificial Life VII, Portland, Oregon, August 2000.

[9] Chapin, J., Markowitz, R., Moxon, K., and Nicolelis, M., "Real-time control of a robot arm using simultaneously recorded neurons in the motor cortex". Nature Neuroscience, Vol.2, Issue.7, pp. 664-670, 1999.

[10] 10.Talwar, S., Xu, S., Hawley, E., Weiss, S., Moxon, K., Chapin, J., "Rat navigation guided by remote control". Nature, Vol. 417, pp. 37-38, 2002.

[11] 11.Denislic, M., Meh, D., "Neurophysiological assessment of peripheral neuropathy in primary Sjögren's syndrome", Journal of Clinical Investigation, Vol. 72, 822-829, 1994.

[12] 12.Poboroniuc, M.S., Fuhr, T., Riener, R., Donaldson, N. "Closed-Loop Control for FES-Supported Standing Up and Sitting Down", Proc. 7th Conf. of the IFESS, Ljubljana, Slovenia, pp. 307-309, 2002.

[13] 13.Popovic, M. R., Keller, T., Moran, M., Dietz, V., 'Neural prosthesis for spinal cord injured subjects', Journal Bioworld, Vol. 1, pp. 6-9, 1998.

[14] 14.Yu, N., Chen, J., Ju, M.; "Closed-Loop Control of Quadriceps/Hamstring activation for FES-Induced Standing-Up Movement of Paraplegics", Journal of Musculoskeletal Research, Vol. 5, No.3, 2001.

[15] 15.Cohen, M., Herder, J. and Martens, W.; "Cyberspatial Audio Technology", JAESJ, J. Acoustical Society of Japan (English), Vol. 20, No. 6, pp. 389-395, November, 1999.

[16] 16.Butz, A., Hollerer, T., Feiner, S., McIntyre, B., Beshers, C. "Enveloping users and computers in a collaborative 3D augmented reality", IWAR99, San Francisco, pp. 35-44, October 20-21, 1999.

[17] 17.Kanda, H., Yogi, T., Ito, Y., Tanaka, S., Watanabe, M and Uchikawa, Y., "Efficient stimulation inducing neural activity in a retinal implant", Proc. IEEE International Conference on Systems, Man and Cybernetics, Vol 4, pp 409-413, 1999.

[18] 18.Kennedy, P., Bakay, R., Moore, M., Adams, K. and Goldwaithe, J., "Direct control of a computer from the human central nervous system", IEEE Transactions on Rehabilitation Engineering, Vol. 8, pp. 198-202, 2000.

[19] 19.Nam, Y., Chang, J.C.., Wheeler, B.C. and Brewer, G.J., "Gold-coated microelectrode array with Thiol linked self-assembled monolayers for engineering neuronal cultures", IEEE Transactions on Biomedical Engineering, Vol.51, No.1, pp.158-165, 2004.

[20] 20.Gasson, M.., Hutt, B., Goodhew, I., Kyberd, P. and Warwick, K; "Bi-directional human machine interface via direct neural connection", Proc. IEEE Workshop on Robot and Human Interactive Communication, Berlin, German, pp. 265-270, Sept 2002.

[21] 21.Branner, A., Stein, R. B. and Normann, E.A., "Selective "Stimulation of a Cat Sciatic Nerve Using an Array of Varying-Length Micro electrodes", Journal of Neurophysiology, Vol. 54, No. 4, pp. 1585-1594, 2001

[22] 22.Warwick, K., Gasson, M., Hutt, B., Goodhew, I., Kyberd, P., Andrews, B, Teddy, P and Shad. A, "The Application of Implant Technology for Cybernetic Systems", Archives of Neurology, Vol.60, No.10, pp. 1369-1373, 2003.

[23] 23.Warwick, K., Gasson, M., Hutt, B., Goodhew, I., Kyberd, K., Schulzrinne, H. and Wu, X., "Thought Communication and Control: A First Step using Radiotelegraphy", IEE Proceedings-Communications, Vol.151, 2004.

[24] 24.Warwick, K., Gasson, M., Hutt, B. and Goodhew, I., "An attempt to extend human sensory capabilities by means of implant technology", International Journal of Human Computer Interaction, Vol.17, 2004.

Motivational System for Human-Robot Interaction

Xiao Huang and Juyang Weng

Department of Computer Science and Engineering, Michigan State University
East Lansing, MI 48824, USA
{huangxi4,weng}@cse.msu.edu

Abstract. Human-Robot Interaction (HRI) has recently drawn increased attention. Robots can not only passively receive information but also actively emit actions. We present a motivational system for human-robot interaction. The motivational system signals the occurrence of salient sensory inputs, modulates the mapping from sensory inputs to action outputs, and evaluates candidate actions. No salient feature is predefined in the motivational system but instead novelty based on experience, which is applicable to any task. Novelty is defined as an innate drive. Reinforcer is integrated with novelty. Thus, the motivational system of a robot can be developed through interactions with trainers. We treat vision-based neck action selection as a behavior guided by the motivational system. The experimental results are consistent with the attention mechanism in human infants.

1 Introduction

Human-Robot Interaction (HRI) has drawn more and more attention from researchers in Human-Computer Interaction (HCI). Autonomous mobile robots can recognize and track a user, understand his verbal commands, and take actions to serve him. As pointed out in [4], a major reason that makes HRI distinctive from traditional HCI is that robots can not only passively receive information from environment but also make decision and actively change the environment.

Motivated by studies of developmental psychology and neuroscience, developmental learning has become an active area in human-robot interaction [10]. The idea is that a task-nonspecific developmental program designed by a human programmer is built into a developmental robot, which develops its cognitive skills through real-time, online interactions with the environment. Since a developmental robot can emit actions, there must be a motivational system to guide its behaviors. Studies in neuroscience [6] shows that generally, motivational/value systems are distributed in the brain. They signal the occurrence of salient sensory inputs, modulate the mapping from sensory inputs to action outputs, and evaluate candidate actions. Computational models of motivational systems are still few. Breazeal [1] implemented a motivational system for robots by defining some "drives," "emotions," and facial expressions in advance. This motivational

N. Sebe et al. (Eds.): HCI/ECCV 2004, LNCS 3058, pp. 17–27, 2004.

system helps robots engage in meaningful bi-directional social interactions with humans. However, this system is predefined, which can not further develop into more mature stages. In [11] a neural motivational system was proposed to guide an animat to find places to satisfy its drives (e.g., food) and to learn the location of a target only when it would reduce the drive. Even though there are some learning mechanisms in the proposed motivational system, it can only conduct immediate learning while delayed reinforcers can not be learned.

Reinforcement learning for robot control and human-robot interaction is not new and has been widely studied [7]. Computational studies of reinforcement often model rewards into a single value, which facilitates understanding and simplifies computation. However, primed sensation (what is predicted by a robot) has been neglected. Reinforcers are typically sparse in time: they are delivered at infrequent spots along the time axis. Novelty from primed sensation is however dense in time, defined at every sensory refresh cycle. We propose a motivational system that integrates novelty and reinforcers to guide the behaviors of a robot.

To demonstrate the working of the motivational system, we chose a challenging behavior domain: visual attention through neck pan actions. Although the degree of freedom of motor actions is only one, the difficulties lie in the task-nonspecific requirement and the highly complex, uncontrolled visual environment. It is known that animals respond differently to stimuli of different novelties and human babies get bored by constant stimuli. The visual attention task has been investigated by computer vision researchers [5] [8]. However, the past work is always task specific, such as defining static salient features based on the specific task in mind. Important salient features for one task are not necessarily important ones for another task. A novel stimulus for one robot at one time is not novel if it is sensed repeatedly by the same robot. Our approach is fundamentally different from these traditional task-specific approaches in that we treat visual attention selection as a behavior guided by a motivational system. The motivational system does not define saliency of features, but instead novelty based on experience. The attention behavior of the robot is further developed through interactions with human trainers. The experimental results are consistent with the attention mechanism in human infants.

In summary, the reported motivational system proposes the following novel ideas: 1) Primed sensation is introduced as a mechanism to support the motivational system. 2) Our work reported here is the first implemented motivational system as far as we know that integrates general novelty and reinforcement. 3) The motivational system is applicable to uncontrolled environments and is not task-specific. 4) The motivational system itself can develop from its innate form into mature stages. In what follows, we first review the architecture of developmental learning. The detailed motivational system is presented in Section 3. The experimental results are reported in Section 4. Finally, we draw our conclusions and discuss about the future work.

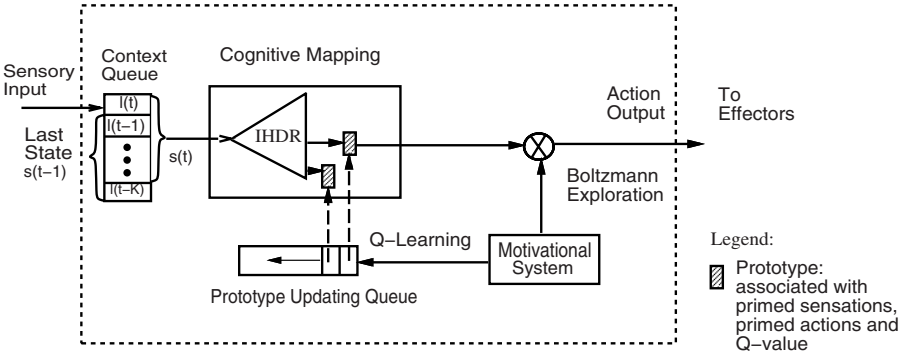


Fig. 1. The system architecture of developmental learning

2 System Architecture

The basic architecture of developmental learning is shown in Fig. 1. The sensory input can be visual, auditory, and tactile, etc, which is represented by a high dimensional vector. The input goes through a context queue and is combined with last state information to generate the current state. Mathematically, this is called observation-driven state transition function $f : S \times L \mapsto S$, where S is the state space, L is the context space. At each time instant, the sensory input $x(t)$ updates the context queue, which includes multiple contexts $l(t)$. $l(t) = \{x(t), p(t), a(t)\}$, that is, a context consists of current sensory input $x(t)$, neck position $p(t)$, and action $a(t)$, where t is the time step. The length of the queue is $K+1$. We should notice that this is a general architecture. We can choose different lengths of the context queue. In the experiment reported here, the length is three. A state $s(t)$ in this experiment consists of two parts: visual image $x(t)$ and neck position $p(t)$. The observation-driven state transition function generates current state from last state and current context $l(t)$, which is defined as $s(t) = f(s(t-1), l(t))$. Last state provides the information of last neck position $p(t-1)$ and last action $a(t-1)$. Based on these two items, we can calculate the current neck position $p(t)$, which is combined with current visual input $x(t)$ to generate the current state $s(t)$. In this case, $s(t)$ covers the information from $l(t-1)$ and $l(t)$ ($K = 2$). It is the cognitive mapping module that maps the current state to the corresponding effector control signal. The cognitive mapping is realized by Incremental Hierarchical Discriminant Regression (IHDR) [3]. A more detailed explanation is beyond scope. Basically, given a state, the IHDR finds the best matched s' associated with a list of primed contexts ($c' = (x', a', q)$), which include: primed sensations $X' = (x'_1, x'_2, ..., x'_n)$, primed actions $A' = (a'_1, a'_2, ..., a'_n)$ and corresponding Q-values $Q = (q_1, q_2, ..., q_n)$, where n is the number of different actions. In other words, the function of IHDR is g: $S \mapsto X' \times A' \times Q$. Primed actions are the possible actions in each state. The probability to take each primed action is based on its Q-value. The primed sensation predicts what will be the actual sensation if the corresponding primed action is taken. The motivational system

works as an action selection function $v : 2^{A'} \mapsto A$ ($2^{A'}$ denotes all the possible subsets of A'), which chooses an action from a list of primed actions.

Novelty is measured by the difference between primed sensation and actual sensation. A novelty-based motivational system is developed into more mature stage through interaction with humans (reward and punishment). Thus, the motivational system can guide a robot in different developmental stages. In order to let the robot explore more states, Boltzmann Softmax exploration is implemented. To reach the requirement of real-time and online updating in developmental learning, we add a prototype updating queue to the architecture, which keeps the most recently visited states (pointed by dash lines). Only states in that queue are updated at each time instant.

3 The Motivational System

The motivational system reported here integrates novelty and reinforcement learning, which provides motivation to a robot and guides its behaviors.

3.1 Novelty

As we know, rewards are sparse in time. In contrast, novelty is defined for every time instant. In order to motivate a developmental robot at any time, it is essential to integrate novelty with rewards. If the ith action is chosen, we can define novelty as the normalized distance between the ith primed sensation $x'_i = (x'_1, x'_2 ... x'_m)$ at time t and the actual sensation $x(t+1)$ at the next time:

$$n(t) = \sqrt{\frac{1}{m} \sum_{j=1}^{m} \frac{(x'_j(t) - x_j(t+1))^2}{\sigma_j^2(t)}} \tag{1}$$

where m is the dimension of sensory input. Each component is divided by the expected deviation σ_j, which is the time-discounted average of the squared difference, as shown in Eq. 2:

$$\sigma_j^2(t) = \frac{t-1-a}{t} \sigma_j^2(t-1) + \frac{1+a}{t} (x'_j(t) - x_j(t))^2 \tag{2}$$

where a is the amnesic parameter to give more weight to the new samples. With an appropriate a, $\sigma(t)$ would represent the short-term variation of the sensation.The amnesic parameter is formulated by Eq. 3:

$$a(t) = \begin{cases} 0 & \text{if } t \leq n_1 \\ c(t-n_1)/(n_2-n_1) & \text{if } n_1 < t \leq n_2 \\ c + (t-n_2)/m & \text{if } n_2 < t \end{cases} \tag{3}$$

where n_1 and n_2 are two switch points, c and m are two constant numbers which determine the shape of a.

3.2 Integration of Novelty and Rewards

However, novelty is only a low level measure. The so defined above just models an innate motivational system. A robot's preference to a sensory input is typically not just a simple function of $n(t)$. Besides novelty, human trainers and the environment can shape the robot's behaviors by issuing rewards and punishments. Furthermore, studies in animal learning show that different reinforcers have different effects. Punishment typically produces a change in behavior much more rapidly than other forms of reinforcers [2]. We integrate novelty and immediate rewards so that the robot can take different factors into account. The combined reward is defined as a weighted sum of physical reinforcers and the novelty:

$$r(t) = w_b r_b(t) + w_g r_g(t) + w_n n(t) \tag{4}$$

where $w_b > w_g > w_n$ are three normalized weights of punishment, reward and novelty respectively, satisfying $w_b + w_g + w_n = 1$.

3.3 Q-Learning Algorithm and Boltzmann Softmax Exploration

However, there are two major problems. First, the reward r is not always consistent. Humans may make mistakes in giving rewards, and thus, the relationship between an action and the actual reward is not always certain. The second is the delayed reward problem. The reward due to an action is typically delayed since the effect of an action is not known until some time after the action is complete. These two problems are dealt with by the following modified Q-learning algorithm. Q-learning is one of the most popular reinforcement learning algorithms [9]. The basic idea is as follows. At each state s, keep a Q-value ($Q(s, c')$) for every possible primed context c'. The primed action associated with the largest value will be selected as output and then a reward $r(t+1)$ will be received. We implemented a modified Q-learning algorithm as follows:

$$\begin{aligned} Q(s(t), c'(t)) \longleftarrow\ & (1-\alpha) Q(s(t), c'(t)) \\ & + \alpha(r(t+1) + \gamma Q(s(t+1), c'(t+1)) \end{aligned} \tag{5}$$

where α and γ are two positive numbers between 0 and 1. $\alpha = (1+a)/t$ is a time varying learning rate based on amnesic average parameter. The parameter γ is for value discount in time. With this algorithm, Q-values are updated according to the immediate reward $r(t+1)$ and the next Q-value. Thus, a delayed reward can be back-propagated in time during learning. The idea of time varying learning rates is derived from human development. In different mature stages, the learning rules of human are different. A single learning rate is not enough. For example, the first time we meet an unknown person, we would remember him right away (high learning rate). Later, when we meet him in different dresses, we would gradually update his image in our brains with lower learning rates. The formulation of α guarantees that it has a large value at the beginning and converges to a constant smaller value through the robot's experience.

We applied the Boltzmann Softmax exploration [7] to the Q-learning algorithm. At each state (s), the robot has a list of primed actions $A(s) = (a'_1, a'_2, ..., a'_n)$ to choose from. The probability for action a to be chosen at s is:

$$p(s, a') = \frac{e^{\frac{Q(s,a')}{\tau}}}{\Sigma_{a' \in A(s)} e^{\frac{Q(s,a')}{\tau}}} \tag{6}$$

where τ is a positive parameter called temperature. With a high temperature, all actions in $A(s)$ have almost the same probability to be chosen. When $\tau \to 0$, the Boltzmann Softmax exploration more likely chooses an action that has a high Q-value. As we know, when we sense a novel stimulus at the first time, we would pay attention to it for a while. In this case, a small τ is preferred because the Q-value of action "stare" would be high and the robot should choose this action. After staring at the novel stimulus for a while, the robot would feel tired and pay attention to other stimuli. Now a larger τ is preferred. After a period of exploration τ should drop again, which means that the state is fully explored and the robot can take the action associated with the highest Q-value. Now the question is how to determine the value of τ. If we choose a large constant τ, then the robot would explore even though it visits a state for the first time. If we choose a small τ, the robot would face the local minimal problem and cannot explore enough states. Fortunately a Guassian density model (Eq. 7) for local temperature solves the dilemma.

$$\tau(t) = \frac{c_1}{(2\pi)^{1/2}\sigma} \exp[-\frac{1}{2}(\frac{t-\mu}{\sigma})^2] + c_2 \tag{7}$$

where c_1 is a constant to control the maximal value of temperature, c_2 controls the minimal value, t is the age of the state, μ and σ are the mean value and standard deviation of the Guassian density model, respectively. The plot of the model can be found in section 4.2. With this model, τ starts as a small value, then climbs to a large value, and finally converges to a small value.

3.4 Prototype Updating Queue

In order to reach the real-time requirement of developmental learning, we designed the prototype updating queue in Fig. 1, which stores the addresses of formerly visited states. Only states in the queue will be updated at each time step. Not only is the Q-value back-propagated, so is the primed sensation. This back-up is performed iteratively from the tail of the queue back to the head of the queue. After the entire queue is updated, the current state's address is pushed into the queue and the oldest state at the head is pushed out of the queue. Because we can limit the length of the queue, real-time updating becomes possible.

3.5 Algorithm of the Motivational System

The algorithm of the motivational system works in the following way:

1. Grab the new sensory input $x(t)$ to update context $l(t)$; combine $l(t)$ with last state $s(t-1)$ to generate current state $s(t)$.
2. Query the IHDR tree and get a matched state s' and related list of primed contexts.
3. If $s(t)$ is significantly different from s', it is considered as a new state and the IHDR tree is updated by saving $s(t)$. Otherwise, use $s(t)$ to update s' through incremental averaging.
4. Update the age of the state, calculate the temperature of the state with Eq. 7.
5. Using the Boltzmann Softmax Exploration in Eq. 6 to choose an action. Execute the action.
6. Calculate novelty with Eq. 1 and integrate with immediate reward $r(t+1)$ with Eq. 4.
7. Update the learning rate based on amnesic average.
8. Update the Q-value of states in PUQ. Go to step 1.

4 Experimental Results

The motivational system is applied to our SAIL robot (short for Self-organizing, Autonomous, Incremental Learner) through vision-guided neck action selection. SAIL, shown in Fig. 2, is a human-size robot at Michigan State University. It has two "eyes," which are controlled by fast pan-tilt heads. In real-time testing, at each step SAIL (placed in a lab) has 3 action choices: turn its neck left, turn its neck right and stay. Totally, there are 7 absolute positions of its neck. Center is position 0, and from left to right is position -3 to 3. Because there is a lot of noise in real-time testing (people come in and come out), we restricted the number of states by applying a Gaussian mask to image input after subtracting the image mean. The dimension of the input image is $30 \times 40 \times 3 \times 2$, where 3 arises from RGB colors and 2 for 2 eyes. The size of the image is 30×40. The state representation consists of visual image and the absolute position of the robot's neck. The two components are normalized so that each has similar weight in the representation. In this experiment, the length of context queue is 3. Biased touch sensors are used to issue punishment (value is set to be -1) and reward (value is set to be 1). The parameters are defined as follows: $\gamma = 0.5$ in Eq. 5; $c = 2$, $n_1 = 1$, $n_2 = 3$, $m = 2$ in Eq. 3. The value of c_1 in Eq. 7 is 5; c_2 is 0.1; μ and σ are 10 and 2, respectively.

4.1 Novelty and Multiple Reinforcers for Different Actions

In order to show the effect of novelty, we allowed the robot to explore by itself for about 5 minutes (200 steps), then kept moving toys at neck position -1. At each position there could be multiple states because the input images at certain neck positions could change. Fig. 3 shows the information of one state at position -1. The image part of the state is the fourth image shown in Fig. 4, which is the background of the experiment. The first three plots are the Q-value of each

Fig. 2. SAIL robot at Michigan State University

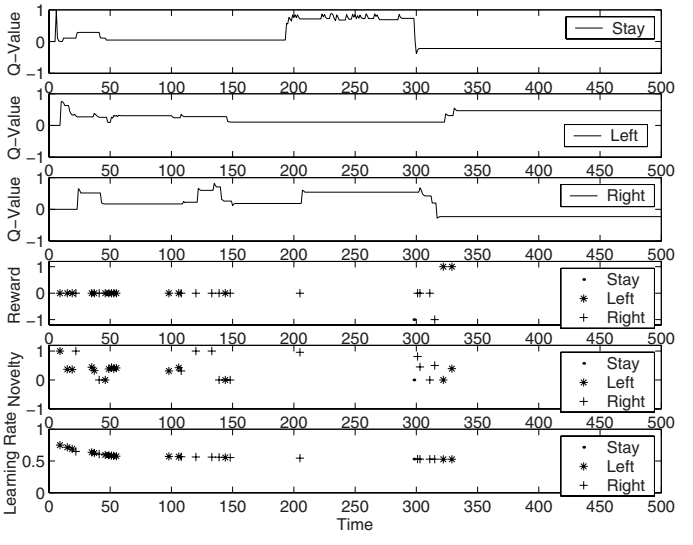


Fig. 3. The Q-value, reward, novelty and learning rate of each action of one state at position 2 when multiple rewards are issued

action (stay, left, right), the fourth plot is the reward of corresponding action, the fifth plot is the novelty value and the last one is the learning rate of the state. After exploration (200 steps later), we moved toys in front of the robot, which increases the novelty and Q-value of action 0 (stay). After training, the robot preferred toys and kept looking at it from step 230 to step 270. A subset of the image sequence is shown in Fig. 4. The first row is images captured by the robot. The second row is the actual visual sensation sequence after applying Gaussian mask. The corresponding primed visual sensation is shown in the third row. If the corresponding sensations in the second the the third row are very different,

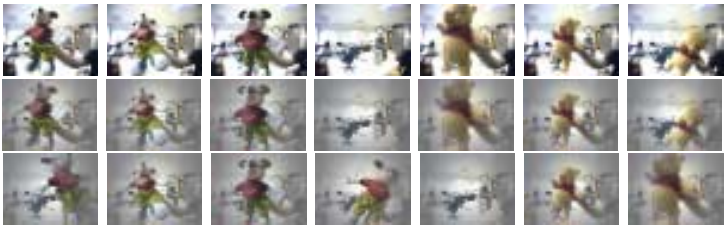


Fig. 4. Part of a sample sensation sequence. The first row is images captured by the robot. The second row is the actual visual sensation input after Gaussian windowing. The corresponding primed visual sensation is shown in the third row

the novelty would be high. The novelty value is shown in the fifth plot. The novelty of action 0, 1, 2 is specified by '.,' '*,' '+,' respectively.

After step 300, the trainers began to issued different reinforcers to different actions. Punishments were issued to action 0 at step 297 and step 298 (the fourth plot) and to action 2 at step 315. Rewards are issued to action 1 at step 322 and step 329. The Q-values of action 0 and action 2 became negative while that of action action 1 became positive, which means that the visual attention ability of the robot is developed through the interactions with the environment. Even though the novelty of action 0 could be high, but the robot will prefer action 1 because of its experience. The learning rate in the fifth row shows that at the beginning the robot immediately remembers the new stimuli and then gradually updates the stimuli.

4.2 Boltzmann Softmax Exploration

As we mentioned in section 3, Boltzmann Softmax exploration is applied so that the robot can experience more states. In Fig. 5, only information from step 1 to step 60 of the above state is shown. The first plot is the probability of each action based on its Q-value. The total probability is 1. The probabilities of action 0, 1, 2 are plotted at the top, middle and bottom, respectively. The star denotes the random value generated by a uniform distribution. If the random value is in one range, say, the middle range, then action 1 would be taken. Because the robot is not always in the state, the plot is kind of sparse. The second plot shows the temperature based on the Gaussian density model of Eq. 7. At the beginning, τ is small and the novelty of the state is high (the initial Q-values of another actions are zero), so the probability of action "stay" is the largest one (almost 100%). The robot would stare at the stimulus for a while. Then, the temperature increases. The probabilities of each action became similar and the robot began to choose other actions and explore more states. After about 10 time steps, the temperature dropped to a small value (0.1) again, the action with larger Q-value would have more chance to be taken.

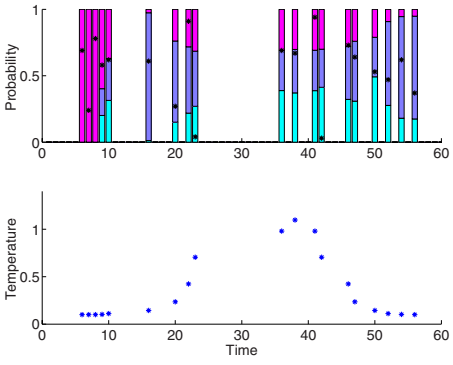


Fig. 5. Boltzmann Softmax Exploration: The total probability is 1. The probabilities of action 0, 1, 2 are plotted at the top, middle and bottom, respectively. The star denotes the random value received at that time. The second plot shows the temperature

5 Conclusions and Future Work

In this paper, a motivational system for human-robot interaction is proposed. Novelty and reinforcement learning are integrated into the motivational system for the first time. The working of the motivational system is shown through vision-based neck action selection. The robot develops its motivational system through its interactions with the world. The robot's behaviors under the guidance of the motivational system are consistent with the attention mechanism in human infants. Since the developmental learning paradigm is a general architecture, we would like to see how the motivational system guides a robot when multiple sensory inputs (vision, speech, touch) are used in human-robot interaction.

References

[1] C. Breazeal. A motivation system for regulating human-robot interaction. In *The Fifteenth National Conference on Artificial Intellilgence*, Madison, WI, 1998. 17
[2] M. Domjan. *The Principles of learning and behavior.* Brooks/Cole Publishing Company, Belmont, CA, 1998. 21
[3] W.S. Hwang and J.J. Weng. Hierarchical discriminat regression. *IEEE Trans. on Patten Analysis and Machine Intelligence*, 22(11):1277–1293, 1999. 19
[4] S. Kiesler and P. Hinds. Introduction to this special issue on human-robot interaction. *Journal of Human-Computer Interaction*, 19(1), 2004. 17
[5] C. Koch and S. Ullman. Shifts in selective visual attention towards the underlying neural circuitry. *Human Neurobiology*, 4:219–227, 1985. 18
[6] O. Sporns. Modeling development and learning in autonomous devices. In *Workshop on Development and Learning*, pages 88–94, E. Lansing, Michigan, USA, April 5–7 2000. 17
[7] R. S. Sutton and A.G. Barto. *Reinforcement Learning – An Introduction.* The MIT Press, Chambridge, MA, 1998. 18, 22

[8] J. Tsotsos, S. Culhane, W. Wai, Y. Lai, N. Davis, and F. Nuflo. Modelling visual attention via selective tuning. *Artificial Intelligence*, 78:507–545, 1995. 18
[9] C.J. Watkins. Q—learning. *Machine Learning*, 8:279–292, 1992. 21
[10] J. Weng, J. McClelland, A. Pentland, O. Sporns, I. Stockman, M. Sur, and E. Thelen. Autonomous mental development by robots and animals. *Science*, 291:599–600, 2000. 17
[11] S. Zrehen and P. Gaussier. Buidling grounded symbols for localization using motivations. In *The fourth European Conference on Artificial Life*, Brighton, UK, July 28-31, 1997. 18

Real-Time Person Tracking and Pointing Gesture Recognition for Human-Robot Interaction

Kai Nickel and Rainer Stiefelhagen

Interactive Systems Laboratories
Universität Karlsruhe (TH), Germany
{nickel,stiefel}@ira.uka.de

Abstract. In this paper, we present our approach for visual tracking of head, hands and head orientation. Given the images provided by a calibrated stereo-camera, color and disparity information are integrated into a multi-hypotheses tracking framework in order to find the 3D-positions of the respective body parts. Based on the hands' motion, an HMM-based approach is applied to recognize pointing gestures. We show experimentally, that the gesture recognition performance can be improved significantly by using visually gained information about head orientation as an additional feature. Our system aims at applications in the field of human-robot interaction, where it is important to do run-on recognition in real-time, to allow for robot's egomotion and not to rely on manual initialization.

1 Introduction

In the upcoming field of household robots, one aspect is of central importance for all kinds of applications that collaborate with humans in a human-centered environment: the ability of the machine for simple, unconstrained and natural interaction with its users. The basis for appropriate robot actions is a comprehensive model of the current surrounding and in particular of the humans involved in interaction. This might require for example the recognition and interpretation of speech, gesture or emotion.

In this paper, we present our current real-time system for visual user modeling. Based on images provided by a stereo-camera, we combine the use of color and disparity information to track the positions of the user's head and hands and to estimate head orientation. Although this is a very basic representation of the human body, we show that it can be used successfully for the recognition of pointing gestures and the estimation of the pointing direction.

A body of literature suggests that people naturally tend to look at the objects with which they interact [1, 2]. In a previous work [3] it turned out, that using information about head orientation can improve accuracy of gesture recognition significantly. That previous evaluation has been conducted using a magnetic sensor. In this paper, we present experiments in pointing gesture recognition using our *visually* gained estimates for head orientation.

N. Sebe et al. (Eds.): HCI/ECCV 2004, LNCS 3058, pp. 28–38, 2004.

Fig. 1. Interaction with the mobile robot. Software components of the robot include: speech recognition, speech synthesis, person and gesture tracking, dialogue management and multimodal fusion of speech and gestures

1.1 Related Work

Visual person tracking is of great importance not only for human-robot-interaction but also for cooperative multi-modal environments or for surveillance applications. There are numerous approaches for the extraction of body features using one or more cameras. In [4], Wren et al. demonstrate the system Pfinder, that uses a statistical model of color and shape to obtain a 2D representation of head and hands. Azarbayejani and Pentland [5] describe a 3D head and hands tracking system that calibrates automatically from watching a moving person. An integrated person tracking approach based on color, dense stereo processing and face pattern detection is proposed by Darrell et al. in [6].

Hidden Markov Models (HMMs) have successfully been applied to the field of gesture recognition. In [7], Starner and Pentland were able to recognize hand gestures out of the vocabulary of the American Sign Language with high accuracy. Becker [8] presents a system for the recognition of Tai Chi gestures based on head and hand tracking. In [9], Wilson and Bobick propose an extension to the HMM framework, that addresses characteristics of parameterized gestures, such as pointing gestures. Jojic et al. [10] describe a method for the estimation of the pointing direction in dense disparity maps.

1.2 Target Scenario: Interaction with a Household Robot

The work presented in this paper is part of our effort to build technologies which aim at enabling natural interaction between humans and robots. In order to communicate naturally with humans, a robot should be able to perceive and interpret all the modalities and cues that humans use during face-to-face communication. These include speech, emotions (facial expressions and tone of voice), gestures, gaze and body language. Furthermore, a robot must be able to perform dialogues with humans, i.e. the robot must understand what the human says or wants and it must be able to give appropriate answers or ask for further clarifications.

We have developed and integrated several components for human-robot interaction with a mobile household robot (see Fig. 1). The target scenario we addressed is a household situation, in which a human can ask the robot questions related to the kitchen (such as "What's in the fridge ?"), ask the robot to set the table, to switch certain lights on or off, to bring certain objects or to obtain suggested recipes from the robot. The current software compontents of the robot include a speech recognizer (user-independent large vocabulary continuous speech), a dialogue component, speech synthesis and the vision-based tracking modules (face- and hand-tracking, gesture recognition, head pose). The vision-based components are used to

- locate and follow the user
- disambiguate objects that were referenced during a dialogue ("Give me *this* cup"). This is done by using both speech and detected pointing gestures in the dialogue model.

2 Tracking Head and Hands

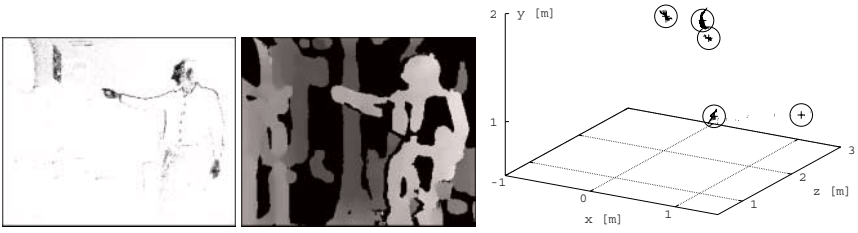
In order to gain information about the location and posture of the person, we track the 3D-positions of the person's head and hands. These trajectories are important features for the recognition of many gestures, including pointing gestures. Our setup consists of a fixed-baseline stereo camera head connected to a standard PC. In our approach we combine color and range information to achieve robust tracking performance In addition to the position of the head, we also measure head orientation using neural networks trained on intensity and disparity images of rotated heads.

2.1 Locating Head and Hands

Head and hands can be identified by color as human skin color clusters in a small region of the chromatic color space [11]. To model the skin-color distribution, two histograms (S^+ and S^-) of color values are built by counting pixels belonging to skin-colored respectively *not*-skin-colored regions in sample images. By means of the histograms, the ratio between $P(S^+|x)$ and $P(S^-|x)$ is calculated for each pixel x of the color image, resulting in a grey-scale map of skin-color probability (Fig. 2.a). To eliminate isolated pixels and to produce closed regions, a combination of morphological operations is applied to the skin-color map.

In order to initialize and maintain the skin-color model automatically, we search for a person's head in the disparity map (Fig. 2.b) of each new frame. Following an approach proposed in [6], we first look for a human-sized connected region, and then check its topmost part for head-like dimensions. Pixels inside the head region contribute to S^+, while all other pixels contribute to S^-. Thus, the skin-color model is continually updated to accommodate changes in light conditions.

In order to find potential *candidates* for the coordinates of head and hands, we search for connected regions in the thresholded skin-color map. For each



a. Skin-color map b. Disparity map c. 3D skin-color clusters

Fig. 2. Feature for locating head and hands. In the skin color map, dark pixels represent high skin-color probability. The disparity map is made up of pixel-wise disparity measurements; the brightness of a pixel corresponds to its distance to the camera. Skin-colored 3D-pixels are clustered using a k-means algorithm. The resulting clusters are depicted by circles

region, we calculate the centroid of the associated 3D-pixels which are weighted by their skin-color probability. If the pixels belonging to one region vary strongly with respect to their distance to the camera, the region is split by applying a k-means clustering method (see Fig. 2.c). We thereby separate objects that are situated on different range levels, but accidentally merged into one object in the 2D-image.

2.2 Single-Hypothesis Tracking

The task of tracking consists in finding the best hypothesis s_t for the positions of head and hands at each time t. The decision is based on the current observation (the 3D skin-pixel clusters) and the hypotheses of the past frames, $s_{t-1}, s_{t-2}, \ldots$.

With each new frame, all combinations of the clusters' centroids are evaluated to find the hypothesis s_t that exhibits the highest results with respect the product of the following 3 scores:

- The *observation score* $P(O_t|s_t)$ is a measure for the extent to which s_t matches the observation O_t. $P(O_t|s_t)$ increases with each pixel that complies with the hypothesis, e.g. a pixel showing strong skin-color at a position the hypothesis predicts to be part of the head.
- The *posture score* $P(s_t)$ is the prior probability of the posture. It is high if the posture represented by s_t is a frequently occurring posture of a human body. It is equal to zero if s_t represents a posture that breaks anatomical constraints. To be able to calculate $P(s_t)$, a model of the human body was built from training data.
- The *transition score* $P(s_t|s_{t-1}, s_{t-2}, \ldots)$ is a measure for the probability of s_t being the successor of the past frame's hypotheses. It is higher, the closer the positions of head and hands in s_t are to the continuation of the path defined by s_{t-1} and s_{t-2}.

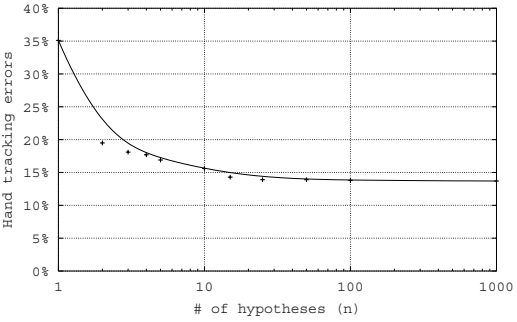


Fig. 3. Percentage of frames with hand-tracking errors in relation to the number of hypotheses per frame (n)

2.3 Multi-Hypotheses Tracking

Accurate tracking of the small, fast moving hands is a hard problem compared to the tracking of the head. The assignment of which hand actually being the left resp. the right hand is especially difficult. Given the assumption, that the right hand will *in general* be observed more often on the right side of the body, the tracker could perform better, if it was able to correct its decision from a future point of view, instead of being tied to a wrong decision it once made.

We implemented multi-hypotheses tracking to allow such kind of rethinking: At each frame, an n-best list of hypotheses is kept, in which each hypothesis is connected to it's predecessor in a tree-like structure. The tracker is free to choose the path, that maximizes overall probability of observation, posture and transition. In order to prevent the tree from becoming too large, we limit both the number n of hypotheses being kept at each frame, as well as the maximum length b of each branch.

2.4 Head Orientation

Our approach for estimating head-orientation is view-based: In each frame, the head's bounding box - as provided by the tracker - is scaled to a size of 24x32 pixels. Two neural networks, one for pan and one for tilt angle, process the head's intensity and disparity image and output the respective rotation angles. The networks we use have a total number of 1597 neurons, organized in 3 layers. They were trained in a person-independent manner on sample images of rotated heads.

2.5 Results

Our experiments indicate that by using the method described, it is possible to track a person robustly, even when the camera is moving and when the background is cluttered. The tracking of the hands is affected by occasional dropouts

and misclassifications. Reasons for this can be temporary occlusions of a hand, a high variance in the visual appearance of hands and the high speed with which people move their hands.

The introduction of multi-hypotheses tracking improves the performance of hand-tracking significantly. Fig. 3 shows the reduction of hand-tracking errors by increasing the number n of hypotheses per frame. In order to detect tracking errors, we labeled head and hand centroids manually. An error is assumed, when the distance of the tracker's hand position to the labeled hand position is higher than 0.15m. Confusing left and right hand therefore counts as two errors.

In our test-set, the mean error of person-independent head orientation estimation was 9.7^o for pan- and 5.6^o for tilt-angle.

3 Recognition of Pointing Gestures

When modeling pointing gestures, we try to model the typical motion pattern of pointing gestures - and not only the static posture of a person during the peak of the gesture. We decompose the gesture into three distinct phases and model each phase with a dedicated HMM. The features used as the models' input are derived from tracking the position of the pointing hand as well as position and orientation of the head.

3.1 Phase Models

When looking at a person performing pointing gestures, one can identify three different phases in the movement of the pointing hand:

- Begin (B): The hand moves from an arbitrary starting position towards the pointing target.
- Hold (H): The hand remains motionless at the pointing position.
- End (E): The hand moves away from the pointing position.

We evaluated pointing gestures performed by 15 different persons, and measured the length of the separate phases (see Table 3.1). Identifying the hold-phase precisely is of great importance for the correct estimation of the pointing direction. However, the hold-phase has the highest variance in duration and can often

Table 1. Average length μ and standard deviation σ of pointing gesture phases. A number of 210 gestures performed by 15 test persons have been evaluated

	μ	σ
Complete gesture	1.75 sec	0.48 sec
Begin	0.52 sec	0.17 sec
Hold	0.72 sec	0.42 sec
End	0.49 sec	0.16 sec

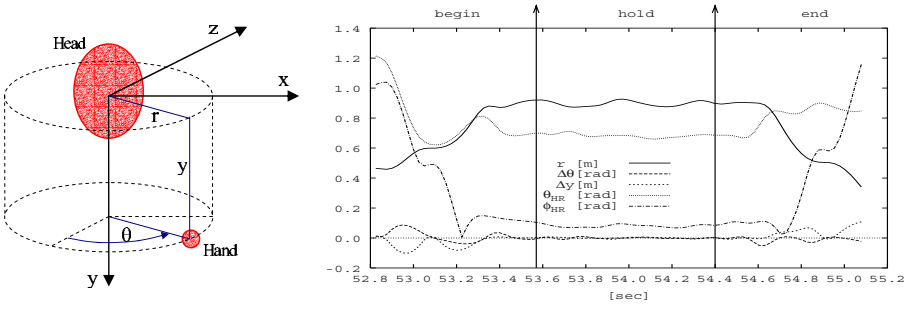


Fig. 4. The hand position is transformed into a cylindrical coordinate system. The plot shows the feature sequence of a typical pointing gesture

be very short (only 0.1sec), thus potentially showing little evidence in an HMM which models the complete gesture. So especially with respect to this fact, we train one dedicated HMM for each of the three phases. In addition to that, there is a null-model, that is trained on sequences that are any hand movements but no pointing gestures.

3.2 Segmentation

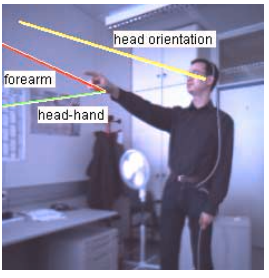
For the task of human-robot interaction we need run-on recognition, meaning that a pointing gesture has to be recognized immediately after it has been performed. So at each frame, we have to search backwards in time for three subsequent feature sequences that have high probabilities of being produced by the begin-/hold-/end-model respectively. The null-model acts as a threshold, such that the phase-models' output must exceed the null-model's output during the course of a gesture. Once a gesture has been detected, its hold-phase is being processed for pointing direction estimation (see section 3.4).

3.3 Features

We evaluated different transformations of the hand position vector, including cartesian, spherical and cylindrical coordinates[1]. In our experiments it turned out that cylindrical coordinates of the hands (see Fig. 4) produce the best results for the pointing task.

The origin of the hands' coordinate system is set to the center of the head, thus we achieve invariance with respect to the person's location. As we want to train only one model to detect both left and right hand gestures, we mirror the left hand to the right hand side by changing the sign of the left hand's x-coordinate. Since the model should not adapt to absolute hand positions – as

[1] See [13] for a comparison of different feature vector transformations for gesture recognition.



	Head-hand line	Forearm line	Head orientation
Avg. error angle	25°	39°	22°
Targets identified	90%	73%	75%
Availability	98%	78%	(100%)

Fig. 5. Comparing different approaches for pointing direction estimation: a) average angle between the extracted pointing line and the ideal line to the target, b) percentage of gestures for which the correct target (1 out of 8) was identified, and c) availability of measurements during the hold-phase

these are determined by the specific pointing targets within the training set – we use the deltas (velocities) of θ and y instead of their absolute values.

In our recorded data, we noticed that people tend to look at pointing targets in the begin- and in the hold-phase of a gesture. This behavior is likely due to the fact that the subjects needed to (visually) find the objects at which they wanted to point. Also, it has been argued before that people generally tend to look at the objects or devices with which they interact (see for example the recent studies in [1] and [2]).

In a previous work [3] it has been shown, that using information about head orientation improves accuracy of gesture recognition significantly. While that evaluation has been conducted using a magnetic sensor, we are now using the visual measurements for head orientation. We calculate the following two features:

$$\theta_{HR} = |\theta_{Head} - \theta_{Hand}| \qquad (1)$$
$$\phi_{HR} = |\phi_{Head} - \phi_{Hand}|$$

θ_{HR} and ϕ_{HR} are defined as the absolute difference between the head's azimuth/elevation angle and the hand's azimuth/elevation angle. Fig. 4 shows a plot of all features values during the course of a typical pointing gesture. As can be seen in the plot, the values of the head-orientation features θ_{HR} and ϕ_{HR} decrease in the begin-phase and increase in the end-phase. In the hold-phase, both values are low, which indicates that the hand is "in line" with head orientation.

3.4 Estimation of the Pointing Direction

We explored three different approaches (see Fig. 5) to estimate the direction of a pointing gesture: 1) the line of sight between head and hand, 2) the orientation of the forearm, and 3) head orientation. While the head and hand positions as well as the forearm orientation were extracted from stereo-images, the head

Table 2. Performance of gesture recognition with and without including head-orientation to the feature vector

	Recall	Precision	Error
Sensor Head-Orientation	78.3%	86.3%	16.8°
Visual Head-Orientation	78.3%	87.1%	16.9°
No Head-Orientation	79.8%	73.6%	19.4°

orientation was measured by means of a magnetic sensor. As we did not want this evaluation to be affected by gesture recognition errors, all gestures have been manually labeled.

The results (see Fig. 5) indicate that most people in our test set intuitively relied on the head-hand line when pointing on a target. This is why we suggest the use of the head-hand line for pointing direction estimation and also use this line in all applications of our system.

3.5 Experiments and Results

In order to evaluate the performance of gesture recognition, we prepared an indoor test scenario with 8 different pointing targets. Test persons were asked to imagine the camera was a household robot. They were to move around within the camera's field of view, every now and then showing the camera one of the marked objects by pointing on it. In total, we captured 129 pointing gestures by 12 subjects.

Our baseline system without head-orientation scored at about 80% recall and 74% precision in gesture recognition (see table 3.5). When head-orientation was added to the feature vector, the results improved significantly in the precision value: the number of false positives could be reduced from about 26% to 13%, while the recall value remained at a similarly high level.

With head-orientation, also the average error in pointing direction estimation was reduced from 19.4° to 16.9°. As the pointing direction estimation is based on the head- and hand-trajectories – which are the same in both cases – the error reduction is the result of the model's increased ability of locating the gesture's hold-phase precisely.

Although there was noise and measurement errors in the visual estimation of head orientation, there was no significant difference in gesture recognition performance between visually and magnetically extracted head-orientation.

4 Conclusion

We have demonstrated a real-time 3D vision system which is able to track a person's head and hands robustly, detect pointing gestures, and to estimate the pointing direction. By following a multi-hypotheses approach in the search for

head and hands, we could improve hand tracking and achieve about 60% relative error reduction.

We could show that the human behavior of looking at the pointing target can be exploited for automatic pointing gesture recognition. By using visual estimates for head orientation as additional features in the gesture model, both the recognition performance and the quality of pointing direction estimation increased significantly. In an experiment (human-robot interaction scenario) we observed a 50% relative reduction of the number of false positives produced by the system and a 13% relative reduction in pointing direction error when using the additional head-orientation features.

Acknowledgements

This work has partially been funded by the European Union under contract nr. FP6-IST-506909 (Project CHIL), and by the German Research Foundation (DFG) as part of the Sonderforschungsbereich 588 "Humanoide Roboter".

References

[1] P. P. Maglio, T. Matlock, C. S. Campbel, S. Zhai, and B. A. Smith. Gaze and speech in attentive user interfaces. *Proceedings of the International Conference on Multimodal Interfaces*, 2000. 28, 35

[2] B. Brumitt, J. Krumm, B. Meyers, and S. Shafer. Let There Be Light: Comparing Interfaces for Homes of the Future. *IEEE Personal Communications*, August 2000. 28, 35

[3] Anonymous. 28, 35

[4] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-Time Tracking of the Human Body. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 7, 1997. 29

[5] A. Azarbayejani and A. Pentland. Real-time self-calibrating stereo person tracking using 3-D shape estimation from blob features. *Proceedings of 13th ICPR*, 1996. 29

[6] T. Darrell, G. Gordon, M. Harville, and J. Woodfill. Integrated person tracking using stereo, color, and pattern detection. *IEEE Conference on Computer Vision and Pattern Recognition*, Santa Barbara, CA, 1998. 29, 30

[7] T. Starner and A. Pentland. Visual Recognition of American Sign Language Using Hidden Markov Models. M. I. T. Media Laboratory, Perceptual Computing Section, Cambridge MA, USA, 1994. 29

[8] D. A. Becker. Sensei: A Real-Time Recognition, Feedback and Training System for T'ai Chi Gestures. M. I. T. Media Lab Perceptual Computing Group Technical Report No. 426, 1997. 29

[9] A. D. Wilson and A. F. Bobick. Recognition and Interpretation of Parametric Gesture. *Intl. Conference on Computer Vision ICCV*, 329-336, 1998. 29

[10] N. Jojic, B. Brumitt, B. Meyers, S. Harris, and T. Huang. Detection and Estimation of Pointing Gestures in Dense Disparity Maps. *IEEE International Conference on Automatic Face and Gesture Recognition*, Grenoble, France, 2000. 29

[11] J. Yang, W. Lu, and A. Waibel. Skin-color modeling and adaption. Technical Report of School of Computer Science, CMU, CMU-CS-97-146, 1997. 30

[12] L. R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proc. IEEE*, 77 (2), 257–286, 1989.

[13] L. W. Campbell, D. A. Becker, A. Azarbayejani, A. F. Bobick, and A. Pentland. Invariant features for 3-D gesture recognition. *Second International Workshop on Face and Gesture Recognition*, Killington VT, 1996. 34

A Vision-Based Gestural Guidance Interface for Mobile Robotic Platforms

Vincent Paquin and Paul Cohen

École Polytechnique de Montréal, Perception and Robotics Laboratory
P.O. Box 6079, Station Centre-Ville
H3C 3A7 Montréal, Canada
{paquin,cohen}@ai.polymtl.ca
http://www.ai.polymtl.ca

Abstract. This paper describes a gestural guidance interface that controls the motion of a mobile platform using a set of predefined static and dynamic hand gestures inspired by the marshalling code. Images captured by an onboard color camera are processed at video rate in order to track the operator's head and hands. The camera pan, tilt and zoom are adjusted by a fuzzy-logic controller so as to track the operator's head and maintain it centered and properly sized within the image plane. Gestural commands are defined as two-hand motion patterns, whose features are provided, at video rate, to a trained neural network. A command is considered recognized once the classifier has produced a series of consistent interpretations. A motion-modifying command is then issued in a way that ensures motion coherence and smoothness. The guidance system can be trained online.

1 Introduction

Gesture-based communications between an operator and a mobile platform constitute a logical step in the search for more natural human-machine interactions. For these interactions to be called natural, the operator must not wear specialized apparatus, such as coloured gloves or electromagnetic tracking devices, and the platform must rely on onboard image interpretation only.

This paper describes a method that uses a standard-resolution colour camera with onboard real-time processing of the captured image sequences. The operator's head is localized with a combination of colour cues and facial features. The hands trajectories around the head are then extracted, filtered and classified as gestural commands. Applications are numerous: machinery manoeuvring, aircraft landing, video games, etc.

Various researchers, who used methods such as background subtraction [2], color segmentation [4] and optical flow computation [3], have reported relative success. Most of the published methods combine the use of a static camera and of somewhat controlled backgrounds, or rely on the assumption that the head and hands are the three largest or the three only skin-like coloured blobs in the image [4, 3, 6].

N. Sebe et al. (Eds.): HCI/ECCV 2004, LNCS 3058, pp. 39–47, 2004.

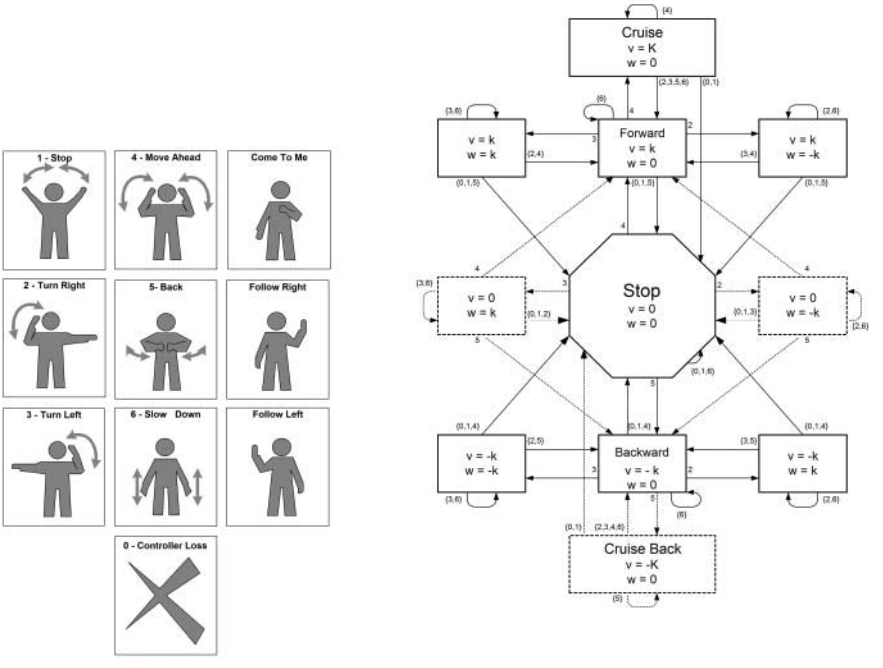
The approach described in this paper uses a colour camera mounted on the mobile platform and works in contexts of unknown operator's surroundings. Images captured by the camera are processed at video rate so as to track the operator's head and hands. A fuzzy-logic controller adjusts the camera pan tilt and zoom, in order to track the operator's head and maintain it centered and properly sized within the image plane. Gestural commands are defined as two-hand motion patterns, whose features are provided, at video rate, to a trained neural network. A command is considered recognized once the classifier has produced a series of consistent interpretations. A motion-modifying command is then issued to the platform in a way that ensures motion coherence and smoothness. The guidance system can be trained online.

An overview of the system is presented in Section 2, together with the gestural commands and their associated platform behaviours. Sections 3 and 4 respectively describe the low-level image processing and the gesture recognition procedure. Section 5 presents the learning method used and Section 6 provides discussion of results currently obtained and an outline of further developments.

2 Gestural Commands and Platform Behaviour

The gesture communication protocol used here is inspired from the marshalling code, as illustrated in Figure 1(a). The gestural commands are defined either by the hand motions or their static position relative to the head. As illustrated, only large amplitude hand motions are considered and finger postures are ignored. The arrows on the figure indicate these motions; the "Back" gesture corresponds to a push-away motion. The six dynamic gestures on the left side are used for low-level commands while the three static gestures on the right side initiate more complex behaviours.

A state machine ensures the platform reacts coherently and smoothly to the commands issued by the operator. This is illustrated in Figure 1(b) where v and w represent translation and rotation speeds respectively. Dashed lines correspond to optional states and transitions. The gesture command protocol can be installed onto any type of mobile platform; however, it is sometimes impossible for a given platform to be in certain states (a car-like steered platform, for example, cannot turn on itself with a null translational speed while a skid-steer platform can). For obvious safety reasons, the "Stop" state should be reached from any other state and the robot must stop if it loses track of the operator's head for an extended period of time. Transitions from forward motion to backward motion or from "Turn Left" to "Turn Right" without passing through an intermediary state are forbidden in order to protect the hardware from brutal transitions. For the sake of clarity, the "Follow Left", "Follow Right" and "Come to me" states are not represented but they can be reached from or reach any other state. Once recognized, the "Follow Right" and "Follow Left" commands trigger platform motions that use a guiding structure (such as a wall) for autonomous motion. The "Come to me" command is interpreted on the basis of parameters extracted from the pan/tilt unit and zoom position.



(a) (b)

Fig. 1. Gestural Commands and State Machine

In order to limit the complexity and enhance the performance of the image interpretation modules, the following constraints are imposed: (1) there must be only one person (the operator) in the camera's field of view; (2) the operator must wear a long-sleeved shirt and face the camera during gesturing; and (3) the operator must not wear skin-coloured garments.

An overview of the software architecture is illustrated in Figure 2 where the dotted lines correspond to future developments. Modules are presented in the following sections.

3 Skin Blob and Face Detection

RGB images are captured at a rate of 30 fps and digitalized at a resolution of 320X240X24 bits. The frames are first filtered using a Gaussian kernel to smooth out irrelevant variations and enhance large blobs. The gradients are computed simultaneously and the image is then divided in 16X16 pixels windows. The sum of the gradients magnitude is taken over each window and compared to a threshold. This allows processing only regions that have enough high frequency content and

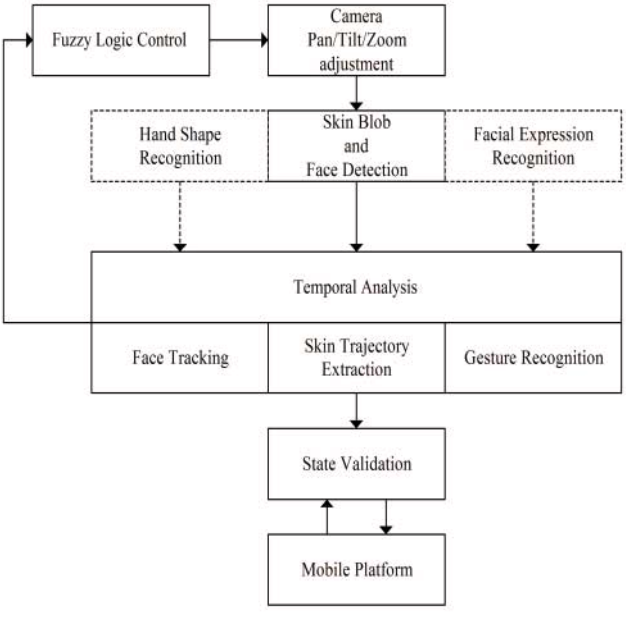


Fig. 2. System Architecture

speeding up later computations. Skin-coloured pixels are then localized in the blurred frames. According to Albiol et al. [1], the performance of an optimal skin detector is independent of the colour space used. However, the computational efficiency requirement limits the performance (and therefore precludes optimality) of the retained representation. The YIQ colour space was chosen over other representations for the speed of conversion from RGB to YIQ and the fact that chromaticity information is limited to a plane (limiting comparisons with skin models to two-dimensional ones). The pixels that fit the model are gathered together in blobs by an optimized flood fill algorithm. Facial and elliptical features are also localized in the non smoothed frames and compared with the results of the blob detection. The face region is defined as a large skin-coloured blob containing facial and/or elliptical features. This approach produces significantly more reliable results than to simply consider the largest blob as a face.

4 Temporal Analysis

The temporal analysis modules handle face tracking, hand trajectory extraction and gesture recognition. Face tracking is straightforward when the operator faces the camera, since the face detector combined with colour cues usually provide reliable results. However, additional computation is needed when the face of the

operator turns away from the camera, due to the platform motion. Eligible blobs are compared with the face blob from the previous image and a reliability score is attributed to each comparison, based upon size, distance and colour content. Whenever the highest reliability score falls under a specified threshold, the head is assumed to be lost until the operator faces the camera again. Once the head is detected in a new frame, its position and size are sent to the fuzzy logic controller to adjust the pan/tilt/zoom values so as to maintain the face's size and center position in the image.

Once the head is detected, other blobs are filtered using geometrical and colour criteria. A blob is dismissed when its mean colour and/or its mean illumination are too far from the head blob's own colour and illumination values or when its size is larger than the head region. Structures of man-made environments that share colour characteristics with the skin model are easily removed since they usually show a lower variance in chromaticity due to their uniform and precise colour (brown wooden doors for example). At time t, the remaining blobs are used to fill the trajectory matrix M in Equation (1), where C_{max} is the counters maximal value, S is the decrement step size and B is a Boolean matrix representing the presence of a skin blob center at position (x, y).

$$M_{x,y}(t) = max(max(C_{max} \cdot B_{x,y}(t), M_{x,y}(t-1)) - S, 0) \tag{1}$$

Normalization maintains the head's width at a fixed size (which experimentation has proved more robust than the height). When the counters $M_{x,y}$ are set to zero, the pixels are erased from the matrix so that each moving skin-coloured blob center seems to be followed by a trail. In a perfect world, there would only be one trail on each side of the head, one for each hand. However, since the background is not controlled, each side generally contains more than one trail. To solve this problem, trajectory matrices are filtered by neural filters associated to each gesture. The left-hand trail is paired up with the right-hand side. Moment-based features are extracted from the pairs and provided to a trained neural network. A gestural command is recognized once the classifier produces enough temporally consistent interpretations.

5 Learning

While a hard-coded classifier could have been used to recognize gestures, a multilayer perceptron was preferred since it could be trained online. The training must to be done in "ideal" conditions, i.e. there must be no skin blobs other than the head and the two hands. During training, features are extracted and stored in a database with their corresponding classes. When a suitable number of examples for each gesture have been collected, the trainer can launch the Levenberg-Marquardt [5] back propagation training algorithm. Since the number of features is low (about 14) and the gestures are quite different from each other in feature space, the amount of necessary examples is very low (about 8 to 12 per gesture) and the training demands little time. The neural filters associated

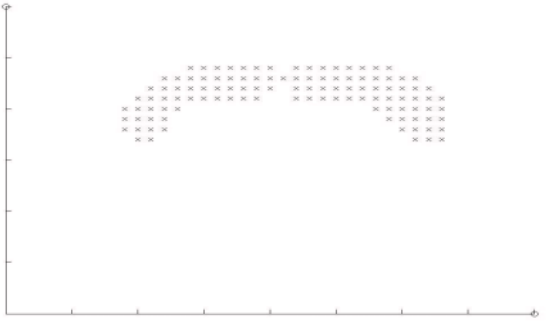


Fig. 3. Stop Filter Output

Fig. 4. Experimentation

with each gesture are trained simultaneously. They are provided with the x, y coordinates of the trajectory pixels and taught to decide whether a pixel is a part of a given trajectory or not. Figure 3 presents the output for the "Stop" command. Further development will allow adding new gestures online without having to modify the code.

6 Experiments and Discussion

The recognition method has been implemented on a laptop equipped with a Pentium 4/2.4GHz processor. The tests where performed on an iRobot's ATRV, as illustrated on Figure 4. The robot equipment includes a Sony EVI-D100 pan/tilt/zoom camera system.

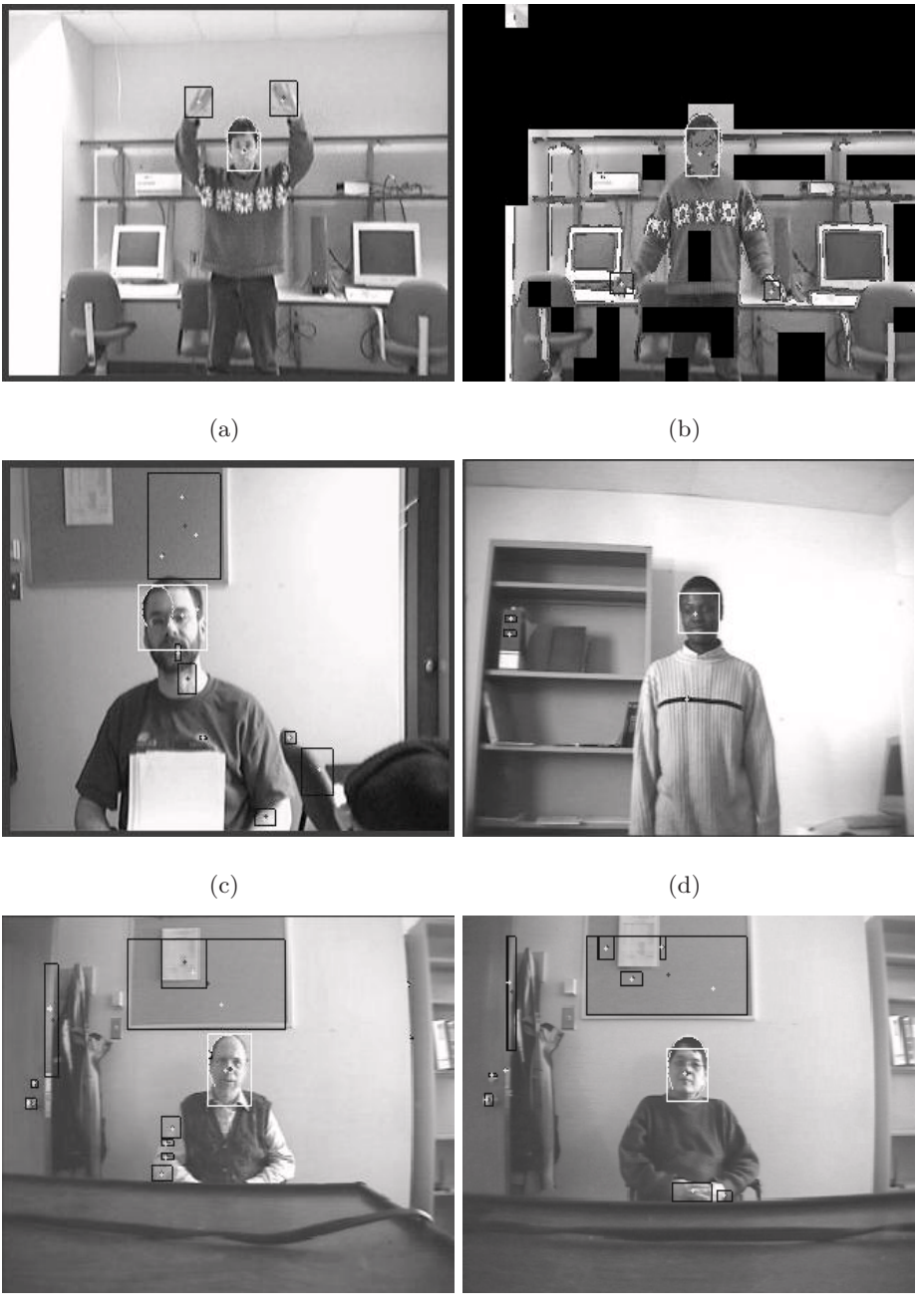
Table 1. Recognition rates

	Noise-to-Signal Ratio		
Gesture	0%	15%	31%
Forward	22/22	20/22	20/22
Backward	16/17	15/17	14/17
Stop	30/30	30/30	27/30
Left	33/34	33/34	31/34
Right	33/34	21/34	17/34
Slow Down	18/18	18/18	15/18
False positive	7.2%	24.0%	35.4%

Experiments were first conducted on a fixed platform, so as to test independently the head detection and tracking capability. Reliable results are usually achieved, except under two adverse conditions: (1) wide lighting variations and (2) operator complexion (dark skin or white beard for example). Head tracking performs well at rates of 30 fps down to 7 fps. Lower frame rates may be of interest in contexts of limited CPU. Figures 5(a), 5(d), 5(c), 5(b), 5(e) and 5(f) present results of head tracker experiments performed on different persons standing in front of controlled and uncontrolled background. In those images, areas where faces are detected are represented by white bounding boxes while the remaining ones correspond to other skin-coloured regions. The tracker performs well, even in the case of a bearded subject. In figure 5(c), the ellipse does not fit well because the head is larger than what the zoom controller allows and the ellipse search is limited in size range. Figure 5(b) also shows the discarded low frequency regions (painted in black).

Image sequences were taken in conditions excluding any skin-coloured objects other than the operator's head and hands, in order to test the second stage performance. Salt-and-pepper random noise was added to the extracted hand trajectories (with noise-to-signal ratios of 15% and 31%, corresponding to the average noise points to real trajectory points ratio). Table 1 presents the results of those experiments. False positives are presented as a ratio to the recognized real gestures.

High recognition rates are achieved, with only a small number of false positives. The "Slow Down" command is the one that presents the highest rate of false positives due to the gesture's similarity to a hand-resting position. That problem is easily resolved by requiring a larger number of consecutive frames before accepting the recognition, at the expense of response time. Other difficulties arise when the background produces skin-coloured blobs of the same shape and at the same place in the images as the meaningful commands. This is because the blob's origin is always uncertain, either caused by a moving hand or else. A possible solution could be the use of deformable 3D templates as in [7] to check for the presence of a hand in the ambiguous regions. The worst case scenario happens when the operator stands in front of a skin-coloured wall: all the filters



(a) (b)

(c) (d)

(e) (f)

Fig. 5. Face tracker results

produce the perfect trajectory and all detectors recognize their gestures. But, since the head is still detected, the system could handle this problem by asking the operator to move away.

Among the method's interesting properties: (1) it can recognize dynamic as well as static gestures and is operator independent; (2) it is very fast since it recognizes the gestural commands as characters with a light multilayer perceptron; (3) training can be done online and is a lot simpler than that of techniques using Hidden Markov Models; (4) since it uses low resolution images, the method is less sensitive to background noise; (5) the combination of facial features, colour cues and elliptical features make head tracking more reliable than other methods. Additional development is under way to combine the method to a voice command recognition module. As shown in Figure 2, future plans include adding facial expression recognition and finger gesture recognition in order to build a more complete interface.

Acknowledgements

This work has been supported by the Natural Science and Engineering Council of Canada (Grant No. RGPIN 3890-01). The authors would like to thank the personnel of École Polytechnique's Perception and Robotics Lab (in particular T. Gerbaud, M. Sadou and J.J. Carriere) for their help during experiments.

References

[1] A. Albiol, L. Torres, E. J. Delp. "Optimum color spaces for skin detection", *Image Processing, 2001. Proceedings. 2001 International Conference on*, Vol. 1, pp. 122-124 ,7-10 Oct. 2001. 42

[2] O. Bernier, D. Collobert. "Head and hands 3D tracking in real time by the EM algorithm", *Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, 2001. Proceedings. IEEE ICCV Workshop on*, pp. 75-81, 13 July 2001 39

[3] R. Cutler, M. Turk. "View-based interpretation of real-time optical flow for gesture recognition", *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on*, pp. 416-421, 14-16 April 1998. 39

[4] M. Ehreumann, T. Lutticke, R. Dillmann. "Dynamic gestures as an input device for directing a mobile platform " *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, Vol. 3, pp. 2596-2601, 2001 39

[5] M. T. Hagan, M. Menhaj. "Training feedforward networks with the Marquardt algorithm", *IEEE Transactions on Neural Networks*, Vol. 5, no. 6, pp. 989-993, 1994. 43

[6] Y. Ming-Hsuan, N. Ahuja. "Extracting gestural motion trajectories ", *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on*, pp. 10-15, 14-16 April 1998 39

[7] N. Shimada, K. Kimura, Y. Shirai, Y. Kuno. "Hand posture estimation by combining 2-D appearance-based and 3-D model-based approaches", *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, Vol. 3, pp. 705-708, 3-7 Sept. 2000 45

Virtual Touch Screen for Mixed Reality

Martin Tosas and Bai Li

School of Computer Science and IT, University of Nottingham
Jubilee Campus, Nottingham, NG8 1PG, UK
mtb@cs.nott.ac.uk

Abstract. Mixed Reality (MR) opens a new dimension for Human Computer Interaction (HCI). Combined with computer vision (CV) techniques, it is possible to create advanced input devices. This paper describes a novel form of HCI for the MR environment that combines CV with MR to allow a MR user to interact with a floating virtual touch screen using their bare hands. The system allows the visualisation of the virtual interfaces and touch screen through a Head Mounted Display (HMD). Visual tracking and interpretation of the user's hand and finger motion allows the detection of key presses on the virtual touch screen. We describe an implementation of this type of interfaces and demonstrate the results through a virtual keypad application.

1 Introduction

Classical interface devices used in the VR/MR environment include mice, keyboards, and joysticks, but these devices do not fully realise the potential of a VR/MR environment. More sophisticated devices such as 3D mice, wands, and data gloves can be used in a 3D environment, however, they are not only expensive but also need cables and sensors to operate, which renders the approach intrusive. The intrusion is more apparent in an MR environment, where the difference in handling interaction with virtual objects using these devices and interaction with real objects using bare hands creates a barrier between the real and the virtual world.

Computer vision techniques are the least intrusive for interaction in a MR environment. In the case of hand-based interaction, some require the use of special gloves or markers attached to the hand to facilitate the interaction. This is less intrusive than the use of data gloves or 3D mice, as these special gloves or markers are lighter and do not need any wiring. However unadorned hand tracking can make it possible for a MR user to interact with the virtual and the real in the same way, by using their bare hands.

In this paper we propose a virtual touch screen interface, operated with the user's bare hands, as a way of HCI for MR environments. In a MR environment it is possible to lay windows, icons, menus, buttons, or other type of GUI controls, etc, on the surface of the virtual touch screen. Users could see these interfaces floating in front of them using a HMD, and could interact with these virtual interfaces using their bare hands. One or many cameras can be used to track the user's hands and interpret their motion so to detect when the user is ‘clicking', or pressing a button on the virtual surface. The paper is organised as follows. Section 2 of this paper, is a brief literature review on some work related to this research. Section 3 presents a realization and

N. Sebe et al. (Eds.): HCI/ECCV 2004, LNCS 3058, pp. 48-59, 2004.

applications of a multi-user virtual touch screen interface. Section 4 shows the results of a virtual touch screen implementation, in the form of a virtual numeric keypad. Section 5 gives some conclusions, and future work directions.

2 Related Work

Of special relevance to this research is the digital-desk concept. The basic idea is to augment a desk by tracking the hands of the user and interpreting what is happening on the work-surface. This is achieved by means of one or two cameras situated on the top of the work-surface. The system presents information on the desk by means of a projector situated on top of the work-surface. For example, the system in [1] tracks the user's fingertip by correlating a template of the fingertip with the image on the desk. The correlation is performed only over the area surrounding the last detected finger tip position. If the system loses the track of the hand, the user has to put his finger on a square situated in one of the corners of the desk to resume tracking. Results of the method are shown in a finger drawing application.

BrightBoard [2] is a system that uses a video camera and audio feedback to enhance the facilities of an ordinary whiteboard, allowing the user to control a computer through simple marks made on the board. The system described in [3] is a 3D hand gesture interface system. It acquires gesture input from two cameras, recognizes three gestures, and tracks the user's hand in the 3D space. Five spatial parameters (position and orientation in 3D) are computed for index finger and the thumb. The capability of the system is demonstrated with some example applications: video game control, piloting of a virtual plane over a terrain, interaction with 3D objects by grasping and moving the objects. Also in [4] is described a system that tracks the user's hand using two cameras, one from the top, and the other from the side. The system allows drawing in 3D space and handling of 3D virtual objects. [5] describes three gesture recognition applications: the first allows the user to paint with a finger on a wall using virtual ink, the second allows the user to control a presentation using hand gestures, and the third allows the user to move virtual items in a brainstorming session.

Using a wearable computer and a head-mounted camera for tracking the hand, [6] is concerned with a system that allows the user to encircle an object, thereby coarsely segmenting the object. The snapshot of the object is then passed on to a recognition engine for identification. [7] introduces the concept of ‘steerable interfaces for pervasive computing spaces'. This type of interfaces can be displayed on a wall or on a surface near the user. A 3D environment designer allows the user to define the geometry of the environment, surfaces where the interface could be displayed, and the positions of cameras and projectors in the environment. The technologies needed to realize this concept include projecting images at will on different surfaces, visually tracking user's head and hand orientations and positions, and to placing sound at different places in the environment.

A system called ARKB [8] is more similar to the research described in this paper, in which the user sees a virtual keyboard lying horizontally through a video-see-through HMD and is allowed to ‘type' on it with both hands. However, the user needs to wear markers on the fingers to allow hand tracking by the two cameras on the

HMD. Using stereo matching, the system detects when one of the markers is inside the volume of a virtual key and considers the key as being pressed. Finally in [9] is described an application that a virtual touch screen can enable - an immersive VR application of Japanese characters writing. The user can write using a virtual keyboard, and interact with their hands over some 3D widgets arranged in a virtual desk. The system also takes speech input. However, a CyberGlove is used to recognize user's hand motion.

This paper describes a new form of HCI for MR - visual tracking and interpretation of the user's hand to allow interaction with virtual interfaces on a virtual touch screen and real objects in the same way by using the hand.

3 Virtual Touch Screen Interfaces

A virtual touch screen will be perceived by MR users as a number of GUI elements floating in front of them. MR users can interact with these elements using their bare hands by ‘clicking' with their fingers, on windows, icons, menus, etc, in the same way as popular GUI like MSWindows could be operated via a real touch screen. Some advantages of this type of interfaces are:

- Can be projected wherever necessary and moved to wherever necessary.
- Can be comparatively cheaper than using screens, keyboards, mouse, or data gloves, for the amount of functionality that such a system could offer.
- Suitable for use in harsh working conditions. For example in outdoors or some other environment potentially harmful for keyboard, mouse, or screen, like dusty, humid, or underwater environments.
- Suitable for use in situations where physical contact with the user is not appropriate. For example in a hospital operating theatre, a doctor might be in the middle of an operation on a patient and need to check for some information, or select other functions of a medical AR system. They could do this by using a virtual touch screen without risk of contaminating the hands.

3.1 A Realisation of Multi-User Virtual Touch Screen Interfaces

This section describes how a multi-user virtual touch screen could be realised in an MR environment. The concepts are explained in the context of several development stages of a virtual keyboard. The choice of a virtual keyboard is a good example of a virtual touch screen use, as the hand tracking component involved in recognizing key presses on a virtual keyboard is also applicable for recognizing other types of finger ‘clicks' on buttons, icons, menu items, etc. The system consists of a HMD to allow the user to see the projection of a floating virtual keyboard, and/or other interface elements like windows, icons, etc; a camera to obtain visual input; the software to track user's hands, detect keystrokes on the virtual keyboard, visualise the keyboard and produce visual or acoustic feedback when keys are pressed. Here it is assumed that a single camera is used, but other arrangements using multiple cameras are also allowed.

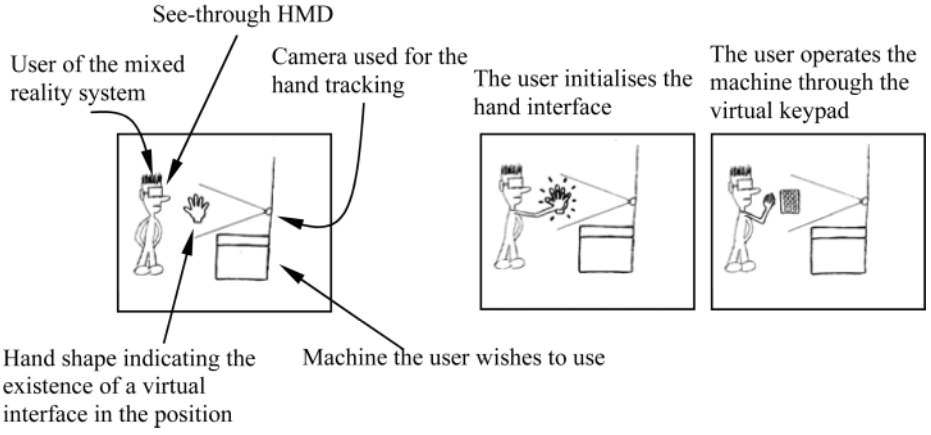


Fig. 1. Initialisation of a static virtual keypad

Fig. 2. Keypad's tilt angle

The easiest implementation would be a *static virtual keypad*. With static here it is meant fixed or tied to some spatial location. The scenario is that the MR user walks towards a machine and sees a projected floating hand shape near the machine. The hand shape indicates that in that position there is a virtual interface ready to be initialised and used. To initialise the virtual interface, the user needs to put his/her hand on top of the projected hand shape. The system then takes a snapshot of the hand and analyses it to initialise hand model parameters for that particular user, such as the lengths of the fingers and finger segments, skin colour, etc. The floating hand shape also sets the initial position for the hand tracking process. If the resolution of the camera is good enough; the snapshot taken from the user's hand could be also used for identification or verification of identity purposes [10]. The hand shape would then disappear and the virtual keypad would appear in its place, ready to be operated by the hand. Visual and acoustic feedback could be produced as the system recognizes the user input, see Fig 1.

The tilt angle of a virtual keypad with reference to the camera could be adjusted for the user's comfort, as can be seen in Fig. 2.

The next stage of development would be to implement a *static alphanumeric virtual keyboard* that could be used with both hands. This is based on the previous keypad idea but in this case hand tracking is slightly more complex as it involves both hands typing at the same time. No restrictions on the hand movement are made to simplify the tracking. Other approaches can be implemented for better results, like different types of keys, different sizes, different tilt angles of the keyboard.

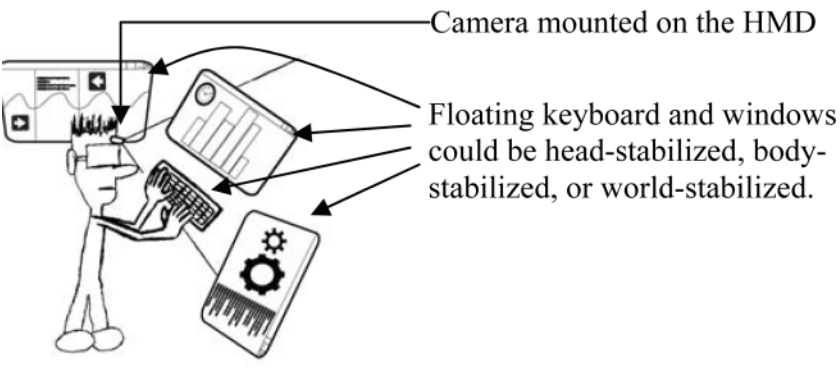


Fig. 3. Floating virtual keyboard and windows

Finally the same concept could be repeated in the form of a *floating virtual keyboard*. In this case the camera for hand tracking is mounted on the user's HMD, in the case of a video see-through HMD this would have already been provided. The keyboard and windows can be presented in three ways in the MR environment: head-stabilized, body-stabilized, and world-stabilized [11], see Fig. 3. However, the user could reposition different virtual elements around themselves by means of some gesture or just ‘clicking' with a finger on some area of a window, keyboard or any other element, or drag the window to a new position.

and tracking in this case is from the back of the hand, which could be more difficult than tracking from the front, as in the case of a static keypad. However if all the interface elements are conceptually placed on the surface of a sphere with the user's head as the centre, then when the hand is actually typing, the view of the hand from the camera is always the same, this can be used to simplify the hand tracking algorithm.

Virtual touch screen interfaces can provide a Windows, Icons, Menus, Pointers (WIMP) interface inside a MR environment. This WIMP interface can be a very useful part of a complex multifunctional MR/AR system. A virtual touch screen can be used to select different functionalities of the MR/AR system. For example, when the user has to manipulate some virtual objects, it will make more sense to use various non-WIMP forms of interaction like speech recognition, gaze tracking, gesture recognition or body/head/arms/hands tracking. However, in a complex system some modalities could interfere with each other. In this case a menu in a virtual touch screen could allow a selection from, for example, hand/arm/body tracking for controlling a robot arm, hand pointing at a 3D location, or gesture recognition to control other type of objects. Ultimately a virtual touch screen could easily present interactive 2D information, texts, and allows the input of alphanumeric information, selection of sequential and discrete items from menus, etc. inside a MR/AR environment.

In Fig. 4 is shown a hypothetical MR user working on the assembling of some virtual objects. The user could select the next object to assemble and see its properties using a virtual touch screen interface and the other hand could control the assembly operation.

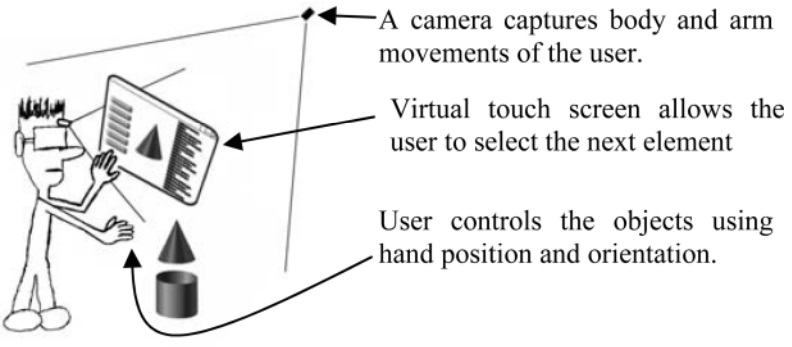


Fig. 4. Virtual touch screen complementing non-WIMP modality for an assembly task

A virtual touch screen would not be suitable for data entry tasks, as the user would have to hold their hand on the air for a long time, resulting in considerable exertion. However it could be suitable for interactive retrieval of information, selection of items from menus, etc. In this case, the user would be clicking, for most of the time, with one of their fingers on some area of the virtual touch screen and only sporadically having to input texts. In addition, the system should allow the user to withdraw their hand from the camera field of view so that the user can have a rest, and resume the hand tracking later.

4 Virtual Keypad Implementation

To illustrate the feasibility of virtual touch screen type of interfaces, a virtual numeric keypad has been implemented. Once the user types on it, the keys pressed will appear on a notepad. The assumptions made here are, the hand is parallel to the camera's image-plane, roughly in vertical position, and against a black background.

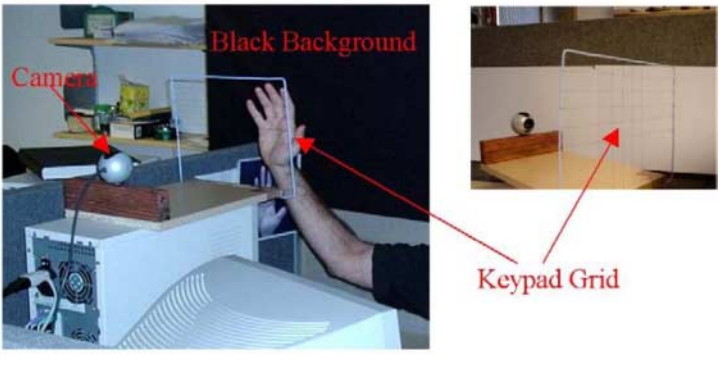


Fig. 5. Set up of the system; camera, keypad grid, and black background

The virtual numeric keypad is intended to be visible through a HMD in an MR application. However at the current stage of this research the main interest is to track

a human hand and interpret the finger movements in an appropriate way so that key presses on a virtual numeric keypad can be detected. The view that provides the HMD has been substituted by a ‘transparent' physical keypad. By transparent it is meant that the user can see the keypad, but tracking can still be performed through it. The first attempt to this ‘transparent' physical keypad was to use a piece of glass standing at a certain distance in front of the camera and parallel to the camera's image plane and draw the numeric keypad on it. In this way the user can see a keypad and attempt to type on it as if it were a real keypad. The hand movements are then recorded and interpreted, and key presses are detected. At the same time on the computer screen a coloured keypad and the tracked hand can be seen so that some feedback is available on the screen, instead of on the HMD. Finally the ‘transparent' keypad is implemented using a frame with some strings forming a grid pattern. The grid serves as the keypad, and the squares forming the grid are the individual keys. For the resolution of the camera, the strings are captured with the width of a single pixel and are later eliminated by an image processing routine. Fig. 5. shows the set up of the system. When the HMD is used instead of the frame and the grid of strings, the MR system will have to display the keypad exactly in the same position as where the grid of strings is.

In the following sections three stages of the virtual keypad operation are described: initial hand tracking, hand model initialisation, and hand tracking with key press detection. As soon as the user puts their hand in front of the keypad, therefore in front of the camera too, the initial hand tracking starts. To initialise the hand model and start using the keypad, the user has to bring their hand, with fingers stretched and hand parallel to the keypad, closer to the keypad up to the point where the palm is touching the strings. Thereafter, hand tracking for key press detection will start.

4.1 Initial Hand Tracking

The initial hand tracking is strictly speaking hand detection. The features detected are the finger-tips, finger-valleys, and hand width. In Fig. 6. the detected finger-tips are marked with blue circles, the finger-valleys are marked with red circles, and the hand width is a horizontal red line that goes from the finger-valley between thumb and index fingers, to the edge of the hand in the other side.

As the background is black the detection of these features is easily performed using basic image processing techniques. The image pixels are analysed in a single pass searching for patterns that look like peaks and valleys. This is followed by a validation step that guarantees that the peaks and valleys found are indeed valid hand features, and not the results of noise, illumination changes, etc. The peaks and valleys found are only counted if they make up five peaks and four valleys, and their relative positions are consistent with a vertical hand. The scheme allows for small rotations of the hand but lager rotations are discarded as they can interfere with the hand tracking for key detection later on. The finger-tip and finger-valley positions will allow in the next two stages, to point out lines longitudinal to the fingers.

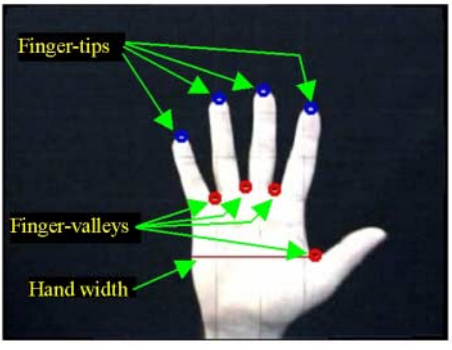


Fig. 6. Finger-tips, Finger-valleys, and Hand width

The hand width is used as a measure of relative distance of the hand to the camera. This approach is taken because it is very simple to implement and is independent of the finger's flexion/extension, unlike other measures like the hand's area, which depends on the distance to the camera and the flexion/extension of the fingers. The distance from the camera to the keypad is known, so when the hand is far away from the keypad, its distance can be approximated comparing the current hand width with the hand width at initialisation time. For this purpose it is important that the hand keeps a vertical orientation at all times.

4.2 Hand Model Initialisation

In section 3.1 it is described how the user has to put their hand on a floating hand shape. The purpose of this step is to take some reference measurements and initialise a hand model. For this particular implementation the hand model consists of the length of the lines longitudinal to the little, ring, middle, and index fingers (called finger lines); and the hand width. Here it is assumed that when the user has the hand inside the floating hand shape, the hand is parallel to the camera's image plane. If the hand model was more complex, then all the necessary reference measurements should be made at this point.

4.3 Key Press Detection

After the hand model is initialised, the hand tracking and key detection operate. When the user's hand goes near enough to the virtual keypad (keypad grid), the keypad displayed on the computer's screen changes colour from red to green, indicating that the hand is on reach to the keypad, ie. the user can type on the keypad from that distance. From that moment on, and while the keypad is green, the lengths of the finger lines are continuously monitored to detect changes in length that can be identified as a key presses. Given one of this length chances (a key press), the final position of the finger tip, before the finger recovers back to the rest position, is checked whether it is inside of the area of a key, in which case the key press of such a key is recognised.

In Fig. 7. is shown an example sequence that involves typing in a telephone number on the virtual keypad. The horizontal axis represents the frame number in the sequence, and the vertical axis represents the length of the finger line. In blue is the length of the Index finger line, in pink is the length of the Middle finger line, and in Yellow is the length of the Ring finger line.

4.4 Test Results

The virtual numeric keypad application was tested with a long typing sequence to check the accuracy of the system. The test involved a sequence containing 121 mostly random key presses. The input to the system was from live video, and the whole sequence was recorded as seen on screen, i.e. showing the virtual keypad, and displaying the finger lines and finger tips/valleys on the hand. After the whole sequence was recorded, a visual inspection of the recorded video allowed verifying how many true-positives, false-negatives and false-positives were produced. The results are presented in Table 1.

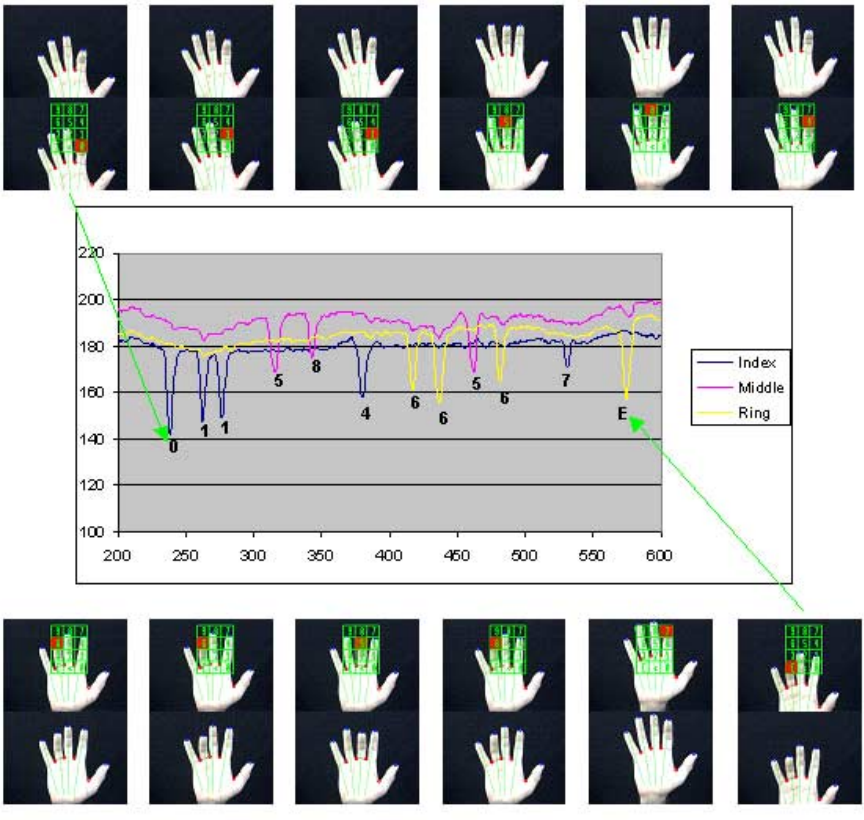


Fig. 7. Typing in a telephone number. In the chart the horizontal axis is the frame number inside the sequence, and the vertical axis is the finger line's length

Table 1. Virtual keypad test results

	Number	Percentage of the total
True-Positives	105	86.7%
False-Positives	4	3.3%
False-Negatives	12	9.91%
Total	121	

The system is slowed down by the recording of the output. This has a negative effect on the accuracy of the system. The recording frame rate had to be set at a trade-off between loss of accuracy and being able to inspect the video properly to identify true-positives, false-negatives and false-positives. The 86.7% accuracy is sufficient for the user to input numeric values quite consistently. In case of the occasional false-negative (a key is pressed but not recorded), the user can press the key again. In the case of a false-positive (a key is not pressed but is recorded as being pressed), the system provides a backspace key for the user to delete the recorded number. With the current approach, the causes of false-negatives and false-positives are due to the following:

- Breakdown in hand tracking or hand is out of the camera field of view.
- Hand is not parallel to the keypad.
- Self-occlusion of fingers, or cluttered background, varying lighting conditions, etc.
- Key press does not produce enough variation in the length of the finger line.
- Movement of other fingers are detected as key presses.

The accuracy of the system allows for rudimentary use. In addition the system provides keys to correct the input. However the accuracy also depends on how good the user is at using this type of keyboard. There is a learning period when the user has to learn to type in a particular way. This is a limitation of the current approach.

5 Conclusion

We have presented a virtual keypad application that illustrates the virtual touch screen interface idea. With the approach used in the virtual keypad key press detection, the results presented are the best possible. There are still some possible improvements following this model, like considering the finger lines to go from the finger tips to a calculated position for the Metacapophalangeal joint, reducing the influence of adduction/abduction movements on the length of the finger line. However the accuracy with which the position of this joint could be calculated on the image is poor using the current hand model. On the other hand, it can be seen that in order to improve the accuracy and relax the constraints of the system, ie. black background, hand parallel to the keypad, no hand rotation, typing in a particular way, etc, the hand model has to be more sophisticated and the tracking approach has to change. The desired requirements of a virtual touch screen are:

- The hand tracking has to be possible in general background conditions, uncontrolled lighting conditions, and views.
- The use of a virtual touch screen should be as natural as possible, for example if the interface on the touch screen is a keyboard, the typing should not need to exaggerate the finger movements.
- The accuracy of the system should be 100%.

Ideally, all these requirements should be met. For this purpose we are planning to use a robust 2D tracking of the hand contour using active contours and deformable template techniques [12, 13]. This 2D tracking allows doing robust tracking of the hand contour, which can be interpreted in combination with a more sophisticated 3D hand model, so that the positions of hand and fingers can be calculated more accurately. The 3D hand model can be a kinematic model of the hand, which would include the lengths of each finger segment and whatever necessary motion constraints. The hand model initialisation stage will have to be able to gather the information needed to initialise the new 3D hand model. At this point many applications can be developed. One example could be to implement a windows manager on a virtual touch screen that can run the same applications as in MSWindows platform, or other type of platforms. This windows manager would need certain adaptations so that texts can be inputted with virtual keyboards.

References

[1] F. Berard J. Crowley and J. Coutaz. Finger tracking as an input device for augmented reality. Proc. Int. Workshop Automatic Face and Gesture Recognition, pages 195-200, 1995.

[2] Peter Robinson Quentin Stafford-Fraser. BrightBoard:a video-augmented environment. Conference proceedings on Human factors in computing systems, ACM Press, pages 134-141, 1996.

[3] Jakub Segen and Senthil Kumar. GestureVR: Vision-based 3d hand interface for spatial interaction. The Sixth ACM International Multimedia Conference, Sep 1998.

[4] Hideo Saito Kiyofumi Abe and Shinji Ozawa. 3-d drawing system via hand motion recognition from two cameras. Proceeding of the 6th Korea-Japan Joint Workshop on Computer Vision, pages 138-143, Jan 2000.

[5] Francois Bérard Christian Von Hardenberg. Bare-hand human computer interaction. Proceedings of the ACM Workshop on Perceptive User Interfaces, 2001.

[6] T. Keaton S. M. Dominguez and A. H. Sayed. Robust finger tracking for wearable computer interfacing. Proc. Workshop on Perspective User Interfaces, Nov. 2001.

[7] Gopal Pingali, Claudio Pinhanez, et al. Steerable interfaces for pervasive computing spaces. First IEEE International Conference on Pervasive Computing and Communications (PerCom'03), page 315, 2003.

[8] Woontack Woo, Minkyung Lee. ARKB: 3d vision-based augmented reality keyboard. ICAT2003(International Conference on Artificial Reality and Telexistence), pages 54-57, Dec, 2003.

[9] Yuji Y. Sugimoto and Noritaka Osawa. Multimodal text input in an immersive environment. ICAT, pages 85-93, December 4-6, 2002.

[10] Landa Silva, Recobos Rodriguez. Biometric identification by dermatoglyphics. Proceedings of the 1996 IEEE International Conference on Image Processing (ICIP 1996), pages 319-322, 1996.
[11] Mark Billinghurst and Hirokazu Kato. Collaborative Mixed Reality. Proceedings of the First International Symposium on Mixed Reality. pages 261-284. 1999.

Typical Sequences Extraction and Recognition*

Gengyu Ma and Xueyin Lin

Key lab on pervasive Computing, Ministry of Education, Computer Science Department
Tsinghua University, Beijing, China, Postcode 100084
Tim99@mails.tsinghua.edu.cn, lxy-dcs@mail.tsinghua.edu.cn

Abstract. This paper presented a temporal sequence analyzing method, aiming at the extraction of typical sequences from an unlabeled dataset. The extraction procedure is based on HMM training and hierarchical separation of WTOM (Weighted Transition Occurring Matrix). During the extraction, HMMs are built each for a kind of typical sequence. Then Threshold Model is used to segment and recognize continuous sequence. The method has been tested on unsupervised event analysis in video surveillance and model learning of athlete actions.

1 Introduction

Recently research on developing new technology of Human Computer Interactive (HCI) has been a hot topic in information technique area, and understanding the human actions, such as facial expression, gesture and body movement etc., automatically by computer has attracted more attention than before. Since human activities are usually the dynamic signal sequences, and different categories of time sequences express different meaning, time sequence clustering is usually the first step for recognizing and understanding human activity. For example, if we want to know the gesture's meaning, its category should be recognized first. In this paper a novel method of extracting typical sequences from a mixture set of sequences is presented. A typical sequence means one kind of sequence which frequently occurs in more and less similar manner, such as typical gestures in human communication and typical activities of a badminton player. Extracting typical sequences automatically can help computer to understand the events and environments. Typical sequences analysis can also be used to detect atypical sequences, which is a hot topic in video surveillance field. It is evident that typical sequence extraction is the problem of data clustering. Therefore the criterion of similarity evaluation between different time sequences should be defined and the clustering strategy should be developed.

A sequence is firstly a dataset of observation follows some distribution. [11] uses cross entropy to measure the distance between two dataset, and [17] uses principal angle to measure the distance. In [17], the method is used to analyze the sequence of people movement and sequence of head tracking images. But this kind of methods

* This research is supported partially by the National Science Foundation of China (No.69975009) and National Key Basic Research and Development Program of China (No.2002CB312101)

N. Sebe et al. (Eds.): HCI/ECCV 2004, LNCS 3058, pp. 60-71, 2004.

ignores the temporal information in the sequences, and just treats the sequence as discrete observations.

Another kind of methods treats sequences as continuous observations. DTW (dynamic time warping) is a method widely used. It measures the distance between two sequences in the clustering method of [2]. But it is only applicable under the condition of small time warping, such as hand writings or curves.

In many complex fields such as speech recognition, HMM has a better performance than DTW [3]. HMM is a stochastic model which describes the probabilities of hidden state transition and observation [4]. Training of HMM, however, needs large amount of data. So it can not be directly used on a single short sequences, such as motion trajectories or image sequences.

HMM can measure the similarity of a sequence respective to a set of sequences. One framework of model building based on HMM adopts the K-mean strategy: First an initial clustering is built, and an HMM is trained for each cluster. Then the cluster membership of each sequence is reassigned based on which cluster's HMM can produce the largest probability. The training and reassigning steps continue until the system parameter converges. The problem of such a framework is that the final result is sensitive to the initial grouping. Most of the researches, adopting such a framework, concentrate on developing a strategy for getting a good initial clustering. Oates, etc. [2], for instance, used DTW in the initial step, and Smyth [5] just randomized the initial clustering.

In our clustering method, however, HMM is used in another way. Similar to [6][7], entropy minimization criterion is adopted for model building. It is well known that entropy can be used to measure the certainty of a model, a compacter model has less uncertainty, and hence a smaller entropy. Entropy minimization is also a theoretical explanation of MDL (minimum description length), which prefer the model that can represent the data with shorter coding length [12, 13]. In [6, 7], entropy minimization is achieved by trimming transitions with small probabilities. In our method, however, entropy reduction is mainly based on separating sequences into clusters. A hierarchical splitting framework is adopted to reduce the interaction between different kinds of sequences. The model building procedure proceeds recursively as follows: A HMM model is trained based on the whole dataset first. Then TOM (Transition Occurring Matrixes) and weighted TOM – WTOM, two features defined in section 2, are calculated for each sequence. Then Normalized Cut [10] is used to split the whole dataset into two clusters. Meanwhile, the entropy of the whole system is reduced. The splitting procedure will continue until all the data in each cluster share the same TOM, and each cluster gains a unique HMM model. A continuous sequence can be segmented into typical and atypical sequences automatically.

The remainder of this paper is organized as follows. The formula of the entropy of an HMM and the structure optimization principle is addressed in section 2. The hierarchical clustering procedure is discussed in section 3. The continuous sequence segmentation and recognition method is presented in section 4. The experiment result, including the clustering of hand gesture, surveillance video, and athlete activity are in section 5. Section 6 is the conclusion.

2 Structure Optimizing Rule

It is well known that entropy can be used to measure the amount of "disorder" of a model, defined as

$$H(p) = \int_{X\in\Omega} -p(X)\ln p(X)dX \tag{1}$$

Where X lies in space Ω, and its distribution is p(X).

2.1 Entropy of a HMM

The parameter of HMM includes three parts: transition matrix $A=(a_{ij})=(P(q_{t+1}=s_j|q_t=s_i))$, which is the transition probabilities among hidden states, prior distribution $\boldsymbol{\pi}=(\pi_i)=(P(q_1=s_i))$, which is the distribution of hidden state at the begin of the sequence, and observation distribution $B=(b_i)=(P(o_t|q_t=s_i))$. Given the HMM parameter, the uncertainty is embedded in its hidden state and observation sequence, so the entropy of it can be calculated as follows.

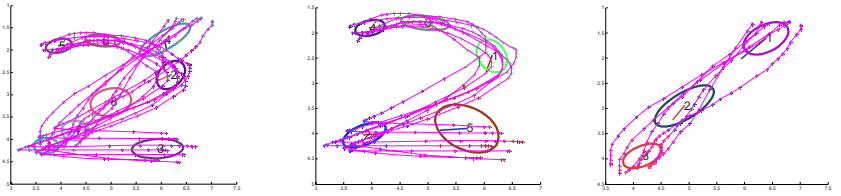
$$\begin{aligned}
&H(P(O,Q))\\
&=-\sum_{q_t\in\{s_1,\ldots,s_L\}}\int_{o_t\in\Omega}P(O,Q)\log P(O,Q)dO\\
&=-\sum_{i=1}^{N}\pi_i\log\pi_i-\sum_{i=1}^{N}\sum_{j=1}^{N}\left(a_{ij}\log a_{ij}\sum_{t=1}^{L}P(q_t=s_i)\right)\\
&-\sum_{t=1}^{L}\sum_{i=1}^{N}\int_{o\in\Omega}P(q_t=s_i)P(o\,|\,q_t=s_i)\log P(o\,|\,q_t=s_i)do\\
&\approx-\sum_{i=1}^{N}\pi_i\log\pi_i-\sum_{i=1}^{N}\sum_{j=1}^{N}\left(t_i a_{ij}\log a_{ij}\right)\\
&-\sum_{i=1}^{N}\left(t_i\int_{o\in\Omega}P(o\,|\,q_t=s_i)\log P(o\,|\,q_t=s_i)do\right)\\
&=H(\pi)+\sum_{i=1}^{N}t_i H(P(q_{t+1}\,|\,q_t=s_i))+\sum_{i=1}^{N}t_i H(P(o\,|\,q_t=s_i))
\end{aligned} \tag{2}$$

The entropy of a HMM is composed of the entropy of its initial distribution, the weighted sum of the entropies of transition probabilities, and the weighted sum of the entropies of observation distributions.

Actually, the entropy of the traditional trained HMM model does not reach the minimum value. In fact, traditional Baum-Welch training method only trains the transition and observation probabilities under given state number and assigned model structure. So the HMM after training is only one of the local minimums. In our method, we train not only the transition probabilities between them, but the hidden state number as well. We train the whole HMM structure, defined as the hidden state number and the transitions between them, according to some rules explained in next section. As a result, the entropy can be reduced towards the global optimum.

2.2 Structure Optimizing Rule

It is well known that HMM is a first order Markov process and hence can not precisely describe complicated situations. Actually high order Markov property can be expressed by dividing a single Markov model into several simpler ones.



(a) (b)

Fig. 1. Express high order Markov process by separated models. (a) trained together; (b) trained separately

For example, in Fig.1, hand writing sequences “1” and “2” are trained together to built a HMM. In the model, hidden state 4 can both transfer to 2 or 8, because the model didn't know where the sequence ends. Evidently this structure does not reflect the possible state transitions properly. Some information lost in this model. But using two models, this information can be represented, Fig.1(b).

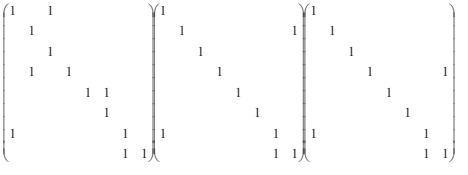
In fact, transition 4→2 conditional depend with the transition 1→3. If a sequence includes 4→2, it must also include 1→3. Therefore uncertainties in the transition of hidden states can be further reduced and hence the entropy. In model (b), the complicated transition path has been explicitly expressed by two branches, and uncertainty is just the prior distribution. In short words, in HMM, the first order Markov assumption can not represent the relations between transitions. Our way of overcoming this is separating the models as shown in (b). Each kind of stroke order has a unique model, so the uncertain transition is removed.

Based on these observations we introduce the structure optimizing rule:

Rule 1: If a transition does not occur in all samples, separate the data into two clusters according to the occurrence of this transition.

To implement this algorithm, we introduce the TOM (Transition Occurring Matrix). For a sequence, if a hidden state transition from i to j occurs, the entry (i,j) in TOM is 1, else it is 0. So rule 1 can be explained as: **all sequences in a cluster have the same TOM.**

In Fig.1(a), there are three kinds of TOMs.



But in Fig1.(b), each cluster has a single TOM.

2.3 Structure Optimization and Entropy

In order to explain the situation mentioned above more clearly, a general situation is shown in Fig.2, in which some sequences pass the hidden states 2-3, while others pass 2-4, and they all pass states other than 2,3, and 4 (‘others' in the Fig.). If the model of

4(a) is split into two models as shown in Fig.2(b), the entropy of either models can be calculated as follows.

Suppose that the prior probabilities of these two types of sequence are (P_1,P_2), the average durations at state 2 are (T_1,T_2), and all parameters besides state 2 are the same for these two kinds of sequences. Remembering that for s=1,2,5,…,N, $t_s=P_1t_{s*}+P_2t_{s'}$, and $t_3=t_{3*}$, $t_4=t_{4'}$ are always satisfied, so the entropies due to the observation are always the same. The only difference of the entropy is due to state 2 and the prior distribution.

In (a), entropy related to the initial distribution is 0, in (b) it is $H([P_1,P_2])= P_1\ln P_1 + P_2\ln P_2$.

The parameters in model (a) can be estimated as

$$a_{23}=\frac{P_1}{P_1\cdot T_1+P_2\cdot T_2},\ a_{24}=\frac{P_1}{P_1\cdot T_1+P_2\cdot T_2},\ \text{and } a_{22}=1-a_{23}-a_{24}.$$

The entropy of the transitions at state 2 in (a) is

$H(P(q_{t+1}|q_t=2)) = H([a_{22}, a_{23}, a_{24}])$.

And the expected occurring time t_2 is $t_2=P_1T_1+P_2T_2$.

In (b) this part of entropy is

$$H(P(q_{t+1}\mid q_t=2^*))=\frac{1}{T_1}\ln\frac{1}{T_1}+\frac{T_1-1}{T_1}\ln\frac{T_1-1}{T_1}$$

$$H(P(q_{t+1}\mid q_t=2'))=\frac{1}{T_2}\ln\frac{1}{T_2}+\frac{T_2-1}{T_2}\ln\frac{T_2-1}{T_2}$$

And the corresponding occurring times are T_1 and T_2 as assumed above.

So the entropies to be compared are:

$$H(a)=(P_1T_1+P_2T_2)*\ln(H(P(q_{t+1}|q_t=2)));$$

$$H(b)=P_1\ln P_1+P_2\ln P_2+T_1P_1H(P(q_{t+1}|q_t=2^*)) +T_2P_2H(P(q_{t+1}|q_t=2'))$$

The functions have 3 variables: T_1, T_2 and P_1. In Fig.3, P_1 is set to 0.5, and the functions are drawn with respect to T_1 and T_2.

For different P_1, the entropy of model (b) is always smaller than or equal to that of model (a). In fact, H(a)=H(b) when $T_1=T_2$. Please notice that in this section only the entropy due to the hidden state transition is concerned, so even the observation distribution is not retrained, this structure modification will reduce the entropy.

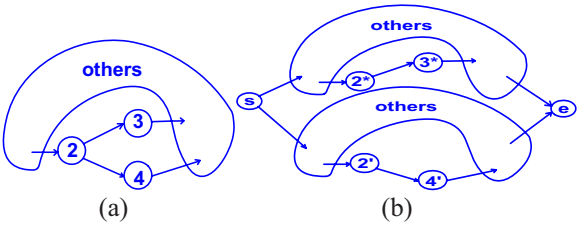
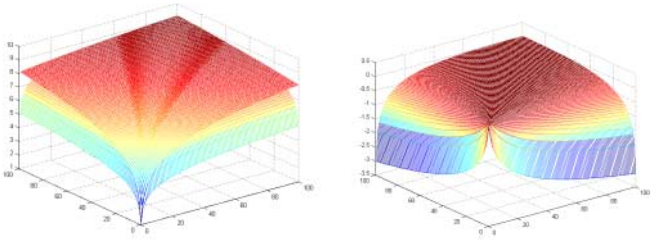


Fig. 2. (a) HMM before modification; (b) HMM after separation



(a) (b)

Fig. 3. (a) The entropies of HMM a and b; (b) H(b)-H(a); Axes x and y are T1 and T2

3 Hierarchical Separations

In our method a hierarchical framework is employed and the mentioned structure optimizing rule is used to split the whole dataset into clusters recursively. The reason of using a hierarchical framework will be explained in section 3.1. Besides, another matrix called weighted Transition Occurring Matrix (WTOM) is defined. The introduced weight is used to evaluate the Gaussianity of each hidden state to make the hierarchical separation procedure more robust.

3.1 Hierarchical Separation

The entire clustering can not directly be based only on the TOMs calculated from the HMM trained from the whole dataset. Since the whole dataset is a mixture of several different kinds of sequences, the cross-interactions between different data and noises prevent this HMM to represent each kind of data well. The handwriting “1” and “2” for example, in the original model, there are totally three kinds of TOMs.

The main advantage of using a hierarchical separation framework is that the corresponding HMM can be updated recursively. By hierarchical separation, the parent dataset are divided into clusters. Data sharing greater similarities are usually assigned within the same cluster. In the process of hierarchical separation, data in each cluster become simpler, their HMM structures and calculated WTOMs become more reliable. Near the leaves of the hierarchical tree, even one difference on TOMs can distinguish two clusters. Using a hierarchical framework, the separation process of handwriting “1” and “2” is as follows. Given the three kinds of TOMs, the TOMs are split into two classes. The difference between TOM 1 and TOM 2 is 4, difference between TOM 2 and TOM 3 is 2, and the difference between TOM 3 and TOM 1 is 4. So the best separation is 1 vs. 23. After this separation, all data are separated into two clusters. For each cluster, a HMM is trained, thus the model in Fig.1(b) is got. In each cluster, all data have the same TOM, so the hierarchical separation ends. Till now, the condition in optimizing rule is fulfilled, so the entropy is minimized.

The hierarchical clustering process is shown in Fig.4.

In the hierarchical separation procedure, a hierarchical tree is built. Each node in the tree corresponds to a cluster and its HMM in the separation process. The root node represents all the training data, and the HMM trained on all the data is called global

HMM. Each node besides the leaves has two child nodes, which are the separation result of this node. In each leaf of the tree, all data have the same TOM. All the clusters embedded in the leaves make up the separation result of the training data.

In the separation procedure TOM is used as the feature of each sequence. Since each TOM is really a pattern, Normalized Cut [10] method can be employed to divide the whole set into two subsets. NCut is a graph cut method. Given a similarity matrix, NCut method finds the separation result with the maximized normalized interclass similarity. In this application, the similarity matrix can be calculated conveniently from the TOMs (and WTOM, which will be introduced later).

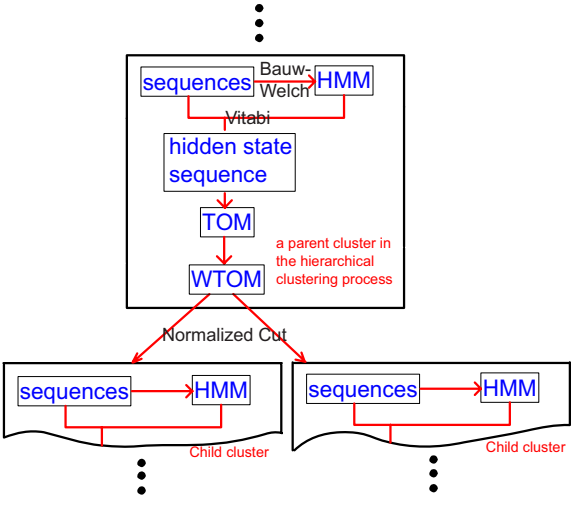


Fig. 4. The hierarchical clustering process

3.2 TOM and WTOM

As mentioned before, TOM is the feature of a sequence that describes its hidden states and their transitions. Since TOMs are calculated from the corresponding HMM, HMM training should be performed first. In this step, the method presented by [16] is employed, which automatically determines the hidden states number of a HMM according to entropy minimization criterion.

Then TOMs are calculated by the Vitabi algorithm. Although TOMs can be used in separation directly, using weighted TOM can make the result more robust. The weight is adopted to judge "whether the transition is good", or "whether the data in the training sample support the distribution". If so the weight value of the corresponding transition will be higher. The weighting value is measured by the evaluation of the Gaussianity of the two states relative to the transition.

Where h_i is the entropy of training data and g_i is the expected entropy of the ideal Gaussian distribution.

For a certain hidden state i, if its observation follows a Gaussian distribution, its entropy will be bigger, so the distribution is more greatly supported by the data. In the WTOM, a state with bigger entropy will has a greater weight value in the computation of similarity matrix.

In the separation process, TOM is used as the stopping criterion, and WTOM is used to calculate similarity matrix used in the NCut algorithm.

$$\begin{aligned} \mathrm{WTOM} &= (t_{ij} w_{ij}) = \left(t_{ij} \exp(h_i - g_i + h_j - g_j)\right) \\ h_i &= -\int P(x) \log P(x) dx \approx -\frac{1}{N} \sum_{t:s_t=i} \log P(o_t \mid \theta_i) \\ g_i &= H(N(\mu, \Sigma)) = \frac{d}{2} + \log((2\pi)^{\frac{d}{2}} \mid \Sigma \mid^{\frac{1}{2}}) \end{aligned} \tag{3}$$

4 Typical Sequence Extraction and Recognition

From the hierarchical separation tree the typical sequences in the training dataset can be obtained. Typical sequence is defined as the kind of sequence that occurs frequently. In the separation result of the training data, a cluster with many samples corresponds to a typical sequence. In the separation procedure, not only typical sequences are found out, for each cluster, the corresponding HMM can also be built. Therefore it can be used to recognize typical sequences from a continuous input sequence.

A continuous sequence usually is a mixture of some typical sequences and some atypical sequences connected together. Therefore some method of recognizing atypical sequence should be included. In our method the entire model is composed of some typical sequence models and a Threshold Model [18]. Each of the first N models is trained by the data in a cluster, and corresponds to a typical sequence. The threshold model corresponds to the atypical sequences. In [18], the Threshold Model is composed of all the hidden states in typical models, their observation probabilities are preserved, but the transitions are set equally, $a_{ij}=a_{ik}=(1-a_{ii})/(N-1)$, $j \neq i$ and $k \neq i$, where N is the number of hidden states. This configuration means no restriction on the transitions between hidden states, so atypical sequence will has a higher probability than that of typical models. In this paper, the atypical sequences are already in the training dataset. So the global model can be directly used as Threshold Model.

While a sequence is given as input, the Vitabi method will find the best suitable hidden state path of the input observation, thus also tells the most probable arrangement of typical sequences and atypical sequences.

5 Experiment Results

The method presented in this paper is a universal method which can be applied on the analysis of many temporal sequences. The hierarchical separation method has been tested on handwriting digits, hand gesture image sequences, human movement, and other sequences. In this section only the experiments on surveillance video and athlete action is demonstrated here for page limitation.

5.1 Typical Event Extraction

The first example demonstrates the usage of our model building method on event extraction from the surveillance video of the entrance lobby of our laboratory, captured by a still surveillance camera. The scene is shown in Fig.5. The entrance is on the left side and a room can be seen inside the door with a phone on the table near it. The lab is beneath the image. The root HMM is shown in Fig.6, and the individual event models are shown in Fig.7, (a) - (i) respectively, with their corresponding meanings. Some abnormal situations are shown in (j) as they seldom occur and can not be recognized as a meaningful cluster.

Sometimes, the atypical sequences should be paid more attention than typical sequences, because atypical sequences are not common events. In this dataset, there are 6 atypical sequences: room → phone → room, room → computer → room, lab → computer → lab, entrance → phone → entry, and two people chatting in the lobby and then go out. The computer runs the surveillance system. And the entry → phone → entry may be produced by a stranger who is not working here. By tradition method, sequences about the computer can be detected because they all passed the ‘computer' state. But the entry → phone → entry sequence is not such lucky, because both the transitions ‘entry → phone' and ‘phone → entry' have high probabilities.

Fig. 5. Surveillance Scene

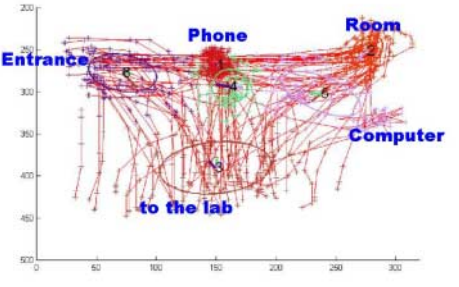


Fig. 6. Parent HMM at the root node of the hierarchical separation tree

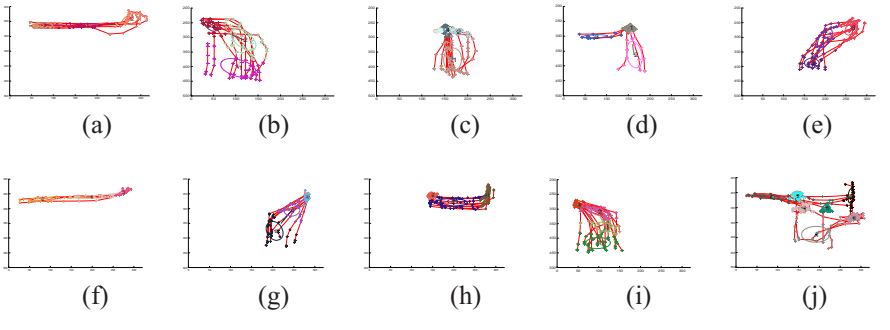


Fig. 7. Extraction result (a) entrance → room; (b) entrance → laboratory; (c) lab → phone → lab (d); lab → phone → entrance; (e) room → lab; (f) room → entrance; (g) lab → room; (h) room → phone → room; (i) lab → entrance; (j) others

5.2 Badminton Player Action Extraction and Recognition

We have used our method for badminton video analysis to extract player's typical actions. In this experiment a badminton video is pre-processed first, so that the player's shape can be segmented from the background. Fig.10 is some action sequences, composed of small player images. Each image was first resized to a normal size; then mapped to a low dimensional space by projecting to a 3D space spanned by the first 3 eigenvectors by means of PCA. Since there are shadows in some of the images, only the upper 2/3 part of an image is concerned. Thus the video is described by some 3D continuous vector sequences. Each sequence corresponds to a shoot in the video, and maybe consists of one or more actions. These long sequences are cut at the points where the athlete shape changes drastically, so that each segment corresponds to a single action.

After the above preprocessing, these action sequences make up the training sample of our sequence clustering algorithm. The training result is shown in Fig.9, each sequence of points represents a typical action. Different kinds of actions are drawn in different color. Each thick line is drawn manually to express the trends of a corresponding typical sequence. To be clear, 9 typical sequences are divided into two groups drawn in two images separately. The x and y axis are the first 2 components in PCA. The corresponding image sequences are shown in Fig.10. From the experimental results it can be seen that, some extracted typical actions, such as smashing, leaning left and leaning right, have explicit semantic meaning. However, there are also some actions that are not so typical to human's opinion. They are usually related to the pre-processing noise. For instance, sequence 1 includes the noise produced by another player's feet; and sequence 4 dues the wrong segmentation.

Finally, we got 9 meaningful actions and integrate them with a threshold model to recognize a continuous input sequence. Fig.9 is the segmentation result. The first frame of each sequence is shown in the figure, the last frame of a sequence just similar to the first frame of the next sequence. The type of each sequence is labeled above each sequence. Some trivial actions are recognized as atypical sequences, labeled as '0'. Six typical sequences are extracted from the video, with one "smashing start" and one "smashing end", two "lean right begin", and two "lean right end" actions. The corresponding frame number is indicated beneath of that image. Actually atypical sequences occupied the main part of the entire video, corresponding to the situation when the player pays attention to his opponent.

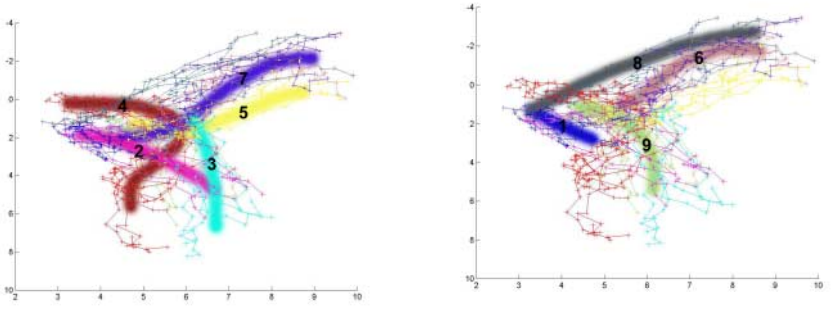


Fig. 8. Typical sequences in the badminton athlete action sequences. (in eigen space)

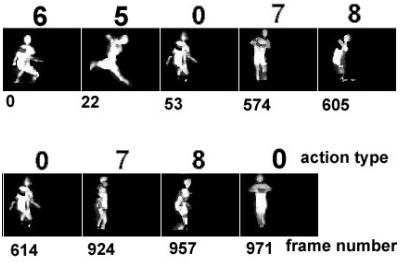


Fig. 9. The segmentation result of another continuous action sequence

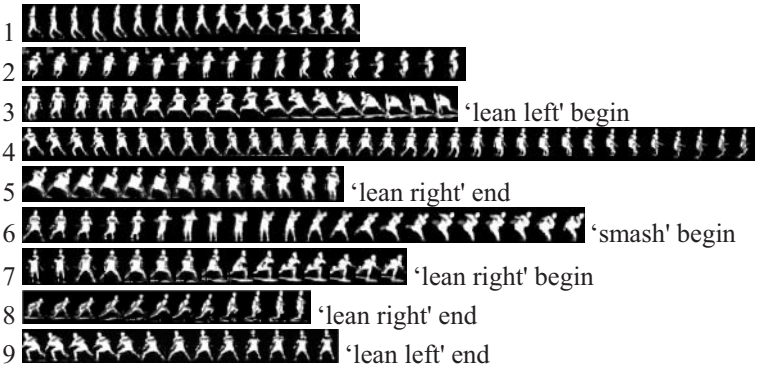


Fig. 10. Typical actions of badminton athlete

6 Conclusion

We proposed a new method to extract typical sequences non-surveillantly. The extraction procedure is a HMM based hierarchical separation procedure. After HMM is trained, TOMs and WTOMs are calculated and used as the features in separation. TOM gives a criterion on when the hierarchical separation process will stop, and WTOM makes the method more robust. Finally the model of the entire system is built by combining the typical models and a Threshold Model. This model is successfully used in the segmentation and recognition of continuous sequences.

References

[1] Berkhin, Pavel.: Survey of Clustering Data Mining Techniques. In: http://citeseer.nj.nec.com/berkhin02survey.html.

[2] Oates, Tim, Laura Firoiu, Paul Cohen.: Clustering Time Series with Hidden Markov Models and Dynamic Time Warping. IJCAI, Working Notes, (1999) 17-21.

[3] Frank Höppner.: Time Series Abstraction Methods - a Survey. Proceedings of GI Jahrestagung Informatik, Workshop on Know- ledge Discovery in Databases, (2002) 777-786,

[4] Awrence R. Rabiner.: A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Proceedings of the IEEE, vol. 77, no. 2, Feb. (1989) 257-285
[5] P. Smyth.: Clustering Sequences with Hidden Markov Models, Advances in Neural Information Processing, MC Mozer, MI Jordan, T. Petsche, Eds. Cambridge, MA, MIT Press, (1997) 648-654
[6] Matthew Brand.: Pattern Discovery via Entropy Minimization. Uncertainty 99 (AI & Statistics) (1999)
[7] Matthew Brand.: An Entropic Estimator for Structure Discovery. NIPS, (1998) 723-729
[8] Matthew Brand, Aaron Hertzmann.: Style machines. SIGGRAPH (2000)
[9] Aapo Hyvarinen and Erki Oja.: Independent Component Analysis: a Tutorial. http://www. cis.hut.fi/~aapo /papers/IJCNN99_tutorialweb/ (1999)
[10] J. Shi and J. Malik.: Normalized Cuts and Image Segmentation. Computer Vision and Pattern Recognition, (1997) 731-738
[11] E. Gokcay, J. Principe.: Information Theoretic Clustering, PAMI, Feb. (2002)
[12] Stine. RA.: Model Selection Using Information Theory and the MDL Principle. http://www-stat.wharton.upenn.edu/~bob/research/smr.pdf
[13] P.M.B. Vitanyi, M. Li.: Ideal MDL and its Relation to Bayesianism. Proc. ISIS: Information, Statistics and Induction in Science World Scientific, (1996) 282-291
[14] Anne Lorette, Xavier Descombes, Josiane Zerubia.: Fully Unsupervised Fuzzy Clustering with Entropy Criterion. ICPR, (2000)
[15] C. Li, G. Biswas.: Improving Clustering with Hidden Markov Models Using Bayesian Model Selection. International Conference on Systems, Man, and Cybernetics, vol. 1, (2000) 194-199
[16] Yoram Singer, Manfred K. Warmth.: Training algorithms for Hidden Markov Models using entropy based distance functions. NIPS, (1996) 641-647
[17] Lior Wolf, Amnon Shashua.: Kernel Principal Angles for Classification Machines with Applications to Image Sequence Interpretation.
[18] Hyeon-Kyu Lee, Jin H.Kim.: An HMM Based Threshold Model Approach for Gesture Recognition, PAMI, Oct. (1999)

Arm-Pointer: 3D Pointing Interface for Real-World Interaction

Eiichi Hosoya, Hidenori Sato, Miki Kitabata
Ikuo Harada, Hisao Nojima, and Akira Onozawa

NTT Microsystem Integration Laboratories
3-1, Morinosato Wakamiya, Atsugi-shi, Kanagawa, 243-0198 Japan
{hosoya,hide,kitabata,harada,nojima,onoz}@aecl.ntt.co.jp

Abstract. We propose a new real-world pointing interface, called Arm-Pointer, for user interaction with real objects. Pointing at objects for which a computer is to perform some operation is a fundamental, yet important, process in human-computer interaction (HCI). Arm-Pointer enables a user to point the computer to a real object directly by extending his arm towards the object. In conventional pointing methods, HCI studies have concentrated on pointing at virtual objects existing in computers. However, there are the vast number of real objects that requires user operation. Arm-Pointer enables users to point at objects in the real world to inform a computer to operate them without the user having to wear any special devices or making direct contacts with the objects. In order to identify the object the user specifies, the position and direction of the arm pointing are recognized by extracting the user's shoulders and arms. Therefore, an easy-to-use real-world oriented interaction system is realized using the proposed method. We developed an algorithm which uses weighted voting for robust recognition. A prototype system using a stereo vision camera was developed and the real-time recognition was confirmed by experiment.

1 Introduction

This paper presents a new method for a real-world pointing system, called Arm-Pointer. Among computer-human interaction techniques, identifying a target object by pointing to the object is one of the most fundamental, yet critical, processes. A typical example of such a pointing process is when we use a computer mouse to point at icons and menus on a computer display.

In a real world environment, there are many objects, such as electronic devices, that are operated by human users. Each real-world object has its own user interface, such as a mouse, keyboard, or remote controller; however, because of the number of such objects, those operations become more complicated. To make the usage of these real-world objects easier, a more intuitively learnable and unified interface is required. From the viewpoint of usability, nothing should be attached to the user's body, the pointing should be detected in real time, and the recognition accuracy should be reliable enough.

Some research have been devoted to achieving object pointing at some distance from the computer display using sensors [1-4]. Put-That-There [1] uses magnetic

N. Sebe et al. (Eds.): HCI/ECCV 2004, LNCS 3058, pp. 72-82, 2004.

sensors and can extract the position of the cursor on the screen. Ubi-Finger [2] is a gesture-input device that uses bending, touch, IR, and acceleration sensors to detect finger gestures for interaction with real objects. These methods can point to a real or virtual object with high accuracy and high speed, but they require the attachment of some equipment to the user [2, 3] or the construction of an environment with sensors [1, 4], which constraints user's position and motion during use.

Other pointing methods based on image processing have been proposed [5-13]. HyperMirror [9] and the Mirror Metaphor Interaction System [10] detect pointing position on a two-dimensional (2D) screen overlapped by self-image. Finger-Pointer [11] and GWINDOWS [12] uses two cameras and extract the three-dimensional (3D) position of the finger to control a cursor on the 2D screen. With these methods, the user does not have to wear any equipment, so they reduce operation-related stress, but they need a display to show the objects to be pointed.

Our proposed method, Arm-Pointer, is a method based on image processing. It can perform 3D recognition of arm pointing direction, which is advantageous for usability and applicability. With Arm-Pointer, the user does not have to wear any equipment and can point to a target real object directly just by stretching out his arm without using a display. Arm-Pointer restricts user position and motion much less than sensor-based methods, and also restricts the shape of user's body much less than other image processing methods that need difficult and robust recognition of the position of small parts or shapes like the finger. Therefore, Arm-Pointer has various applications, such as the control of electric appliances. This paper describes the prototype and discusses its operation and performance based on experimental results.

2 Arm-Pointer

Our goal is to build a real-world pointing system that a user can operate without having to wear equipment and that allows the user to point at object directly in the real world. It is also important for the system to be robust to background images and the user's dress. In what follows, the Arm-Pointer algorithm is explained.

2.1 System Configuration

The configuration of Arm-Pointer is shown in Fig. 1. The system consists of a stereo vision camera, an infrared (IR) remote controller, and a PC. A user stands in front of the stereo vision camera facing it. The user points to a target object by extending his arm, and the system detects that object using the position pointed to. And if the object has a function for interactive operation, the function will be activated. For example, when the user just points at the TV, the TV will turn on.

2.2 Processing Flow

Figure 2 shows the processing/data flow of the whole system. A depth image and a color input image are generated from an output image of a stereo vision camera. A mirror image is also generated and displayed, but this is an additional function for the

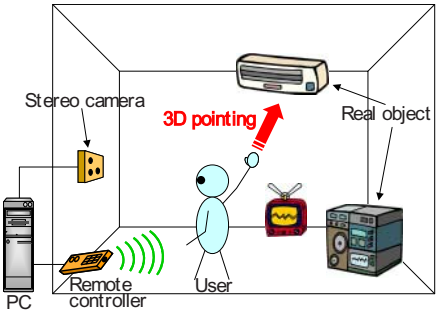


Fig. 1. Arm Pointer: Real-world pointing interface

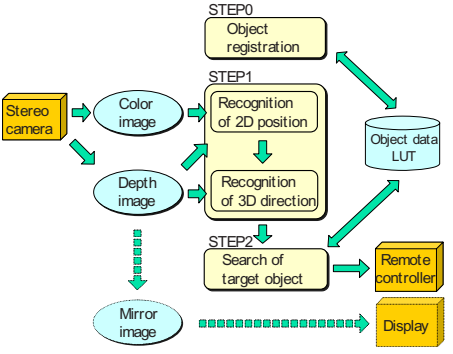


Fig. 2. Processing/data flow

evaluation of user feedback. The system can perform all operations without the display. In the case of interaction with an object, there are roughly two recognition steps. In STEP 1, 3D coordinates of shoulders and arms are recognized and the 3D coordinates of the pointed position are calculated. In STEP 2, the target object is found using the detected position information. All objects are registered beforehand for initialization, which is in STEP 0. STEP 0 to STEP 2 are outlined as follows:

STEP 0 Initialization
: Subdivide voxels and register target objects.

Repeat following steps for each video frame.

STEP 1 3D recognition
: Extract skin color and recognize 3D position of arms and shoulders.

STEP 2 Object search
: Calculate 3D pointing line, vote, and activate command.

3 Initialization

Information about all target objects in the room has to be registered beforehand. This information is used for object search and detection from the pointing direction. In this section, voxel space is defined and the registration of objects described.

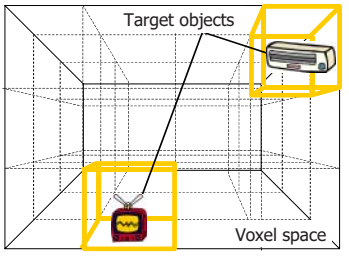


Fig. 3. Object registration in the voxel space

3.1 Voxel Subdivision

Real space is divided into voxel space (discrete space) as shown in Fig. 3. The resolution of voxel space changes with application or the number of objects. Our system can be used in rooms of various sizes by changing the size of voxel space.

3.2 Target Object Registration

We create a look-up table (LUT) corresponding to all voxels. Every record of the LUT contains information about a voxel, such as 3D coordinates, name, and functions. Using this LUT, the system can retrieve the information about a target object, such as televisions, air-conditioners, or walls, by using the 3D coordinates as a search key.

4 3D Recognition and Object Search

The 3D coordinates of the position of the user's shoulders and arms, that is, the 3D pointing direction, are computed by using a depth image and color input image obtained from the stereo vision camera. The object pointed to is detected by determining the pointing direction in the 3D voxel space. First, in this section, the image processing for 3D recognition of 3D pointing direction and the simple object search are described for easy explanation. Next, the improved method is described.

4.1 3D Recognition Flow

The 3D pointing direction is obtained by extracting the 3D position of shoulders and arms, and the 3D position of shoulders are calculated from the 3D position of the face. Fig. 4 shows the coordinate corresponding to each position to be extracted. The steps in the 3D recognition process are outlined as follows:

STEP 1 3D recognition
- STEP 1-1 Extract skin color from input image
  - $h_1<h<h_2$, $s_1<s<s_2$, $v_1<v<v_2$
- STEP 1-2 Recognize face and arms
  - Reduce image noise by 2D image processing
  - Exclude background noise by depth image
  - Exclude error extraction by 2D position restriction of face and arms
  - Calculate 2D coordinates of center of gravity of face and arms; (x_f, y_f), (x_{ar}, y_{ar}), (x_{al}, y_{al})
- STEP1-3 Extract shoulders
  - Calculate 2D coordinates of shoulders from face position;(x_{sr}, y_{sr}), (x_{sl}, y_{sl})
- STEP1-4 Extract distance
  - Calculate 3D coordinates of shoulders and arms from depth image; (x_{sr}, y_{sr}, z_{sr}), (x_{sl}, y_{sl}, z_{sl}), (x_{ar}, y_{ar}, z_{ar}), (x_{al}, y_{al}, z_{al})

In STEP 1-1, the skin color region in the color input image is extracted. The range on the HSV color space of the input image is specified for skin color extraction. The calibration for skin color restrictions is performed beforehand. Each range ($h_1<h<h_2$, $s_1<s<s_2$, $v_1<v<v_2$) is decided by the calibration for the current light condition.

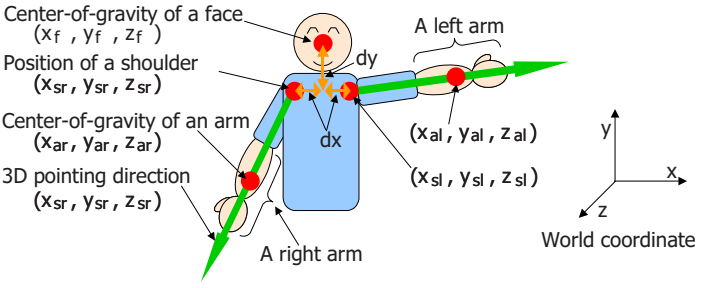


Fig. 4. Calculation for 3D recognition of arms

In STEP 1-2, face and arms recognition is performed. A pre-process reduces noise by avoiding holes and too small areas. The face and arm regions are detected within the given permissible range of user's position, as far as the user faces the camera. With these restrictions, incorrect recognition because of background colors is reduced. Here, (x_f, y_f) are the center-of-gravity coordinates of the face, (x_{ar}, y_{ar}) are the center-of-gravity coordinates of the right arm, and (x_{al}, y_{al}) those of the left arm.

In STEP 1-3, the positions of both shoulders are calculated from (x_f, y_f). Here, for ease of computation, we assume that the user is looking to the front (in the direction of the camera), and that the shoulder is fixed at a certain distance and direction. The coordinates of shoulders, (x_{sr}, y_{sr}) and (x_{sl}, y_{sl}), are determined as those separated certain amount of length from (x_f, y_f).

$$\left.\begin{array}{l} x_{sr} = x_f - dx, \quad y_{sr} = y_f + dy, \quad x_{sl} = x_f + dx, \quad y_{sl} = y_f + dy, \\ dx, dy: \text{difference lengths of between face and shoulders.} \end{array}\right\} \quad (1)$$

Thus, 2D coordinates of the shoulders and arms in a pointing direction can be acquired from a color input image.

In STEP 1-4, the z coordinates of shoulders, z_{sr}, z_{sl}, and arms, z_{ar}, z_{al}, are obtained as depths at their x and y coordinates in the depth image.

The user's 3D pointing direction is indicated by a 3D pointing line that extends from the shoulder along the arm. With this method, the arm has to be fully extended for pointing, but shape recognition of a finger or hand is not necessary. Therefore, there are no constraints imposed by the user's clothes or finger shape.

4.2 Object Search

The target object is searched for in the pointing direction obtained by 3D recognition of the shoulders and arms. The steps in object search are outlined as follows.

STEP 2 Object search
  STEP 2-1 Calculate 3D pointing line
  STEP 2-2 Vote
    Calculate crossing voxel of the line
    If an object is registered in a voxel, increase voting value for the voxel
    If a voting value is over the threshold, go to STEP 2-3, else go to STEP 2-4
  STEP 2-3 Activate command

If the registered object has an interactive function, a command is activated.

STEP 2-4 Decrease voting value after the certain number of frames

In STEP 2-1, the equation of the 3D pointing line is as follows:

$$\frac{x - x_{sr}}{x_{ar} - x_{sr}} = \frac{y - y_{sr}}{y_{ar} - y_{sr}} = \frac{z - z_{sr}}{z_{ar} - z_{sr}} = t_r, \quad (2)$$

$$\frac{x - x_{sl}}{x_{al} - x_{sl}} = \frac{y - y_{sl}}{y_{al} - y_{sl}} = \frac{z - z_{sl}}{z_{al} - z_{sl}} = t_l. \quad (3)$$

In STEP 2-2, voting is performed for detection of the voxel with the target object. First, each voxel crossing the pointing line is detected from the nearest to the farthest in order. If the registered object exists in the voxel, voting value is incremented. If the voting value reaches the threshold after several repetitions for frames, the system goes to STEP 2-3. STEP 2 is repeated until the object is found or room wall is reached. In STEP 2-3, if the detected object is registered as an object with an interactive function, the command of the function is activated. For example, a TV can be turned on or off. In STEP 2-4, the voting value is decreased after a certain time (number of frames).

4.3 Improvements

The method explained in the previous section, referred to as the simple method here, often can not detect a target object quickly. In this section, we propose some improvements using a weighted voting algorithm for better extraction performance. In the improved method, the system votes not only for the crossing voxel through which the pointing line passes, but also voxels neighboring the crossing voxel with a weighted voting value. The area for voting comprises 26 neighboring voxels as shown in Fig. 5. Voting values are decreasing with the distance from the arm to crossing voxel. Using this improved method, it is possible to detect the target object faster, even if the detected position of face and arms are unstable. In this work, we tested Arm-Pointer using only one set of voting weights, which we decided voluntarily. In future work, we will test and evaluate the system using other sets of voting weights to further improve the performance of the voting method. The processing flow in the new voting step (STEP2-2') is as follows. This step replaces STEP2-2 in the simple method.

STEP2-2' Vote

Calculate the crossing voxel V_c (X_c, Y_c, Z_c) that includes an object.
Vote V_c and its 26 neighboring voxels according to the weight,
if they include any object.

5 Application Prototype

We developed a prototype system using our proposed method. The system consists of a PC (Pentium IV, 2 GHz), stereo vision camera (Digiclops), infrared multi-remote controller (CROSSAM2+USB), and image processing software. In this system,

a display can be used additionally for confirming operations by overlaying CG information, although our method can be implemented without a display.

The user has to face the camera and extend his arm to the target. If the object included in the voxel is interactive, its function will be activated. This system makes it possible to operate many apparatuses in the room with only one remote controller.

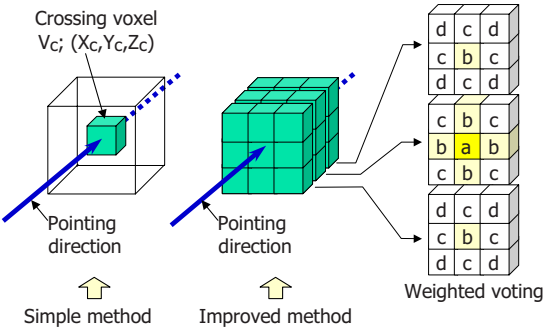
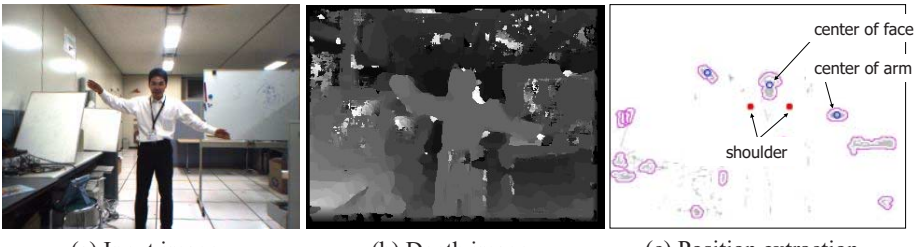
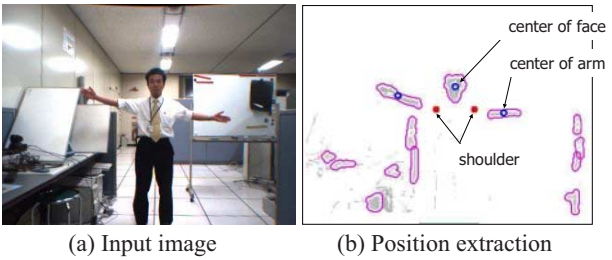


Fig. 5. Weighted voting



(a) Input image (b) Depth image (c) Position extraction

Fig. 6. Examples of 2D extraction



(a) Input image (b) Position extraction

Fig. 7. In the case of short-sleeved shirt

6 Evaluation

We tested the system for evaluation. In this section, examples of user operation and results of a comparison evaluation are shown.

6.1 Experiment

An experiment using the prototype system confirmed basic operations. The processing speed of this system reached about 150 msec/frame, including indication on the display.

An example of the extraction results for the shoulders and centers of the arms are shown in Fig. 6. Figure 6(a) and (b) are an input color image and a depth image, respectively. Figure 6(c) shows the result of skin color extraction, and the 2D position of the center of a face, shoulders, and arms. Although many candidates are extracted at first, finally only the face and arms are detected. Figure 7 shows an example of the extraction results when the user is wearing a short-sleeved shirt. In the case of either a long-sleeved shirt (Fig. 6) or short-sleeved shirt, the 3D direction of the arm can be extracted similarly, without changing any system parameters.

Figure 8 shows an example of the results of the 3D pointing experiment. The red line segment that shows the pointing direction and the voxel frame of the 3D pointing position is displayed and changes. Figure 9 shows an example where interaction with an object was performed. In the case of this figure, since a TV is selected by pointing, the frame of the voxel with the TV is emphasized on the screen. The TV is thus turned on and the message indicated on the screen.

Fig. 8. Examples of 3D pointing

6.2 Comparison Evaluation

For the evaluation, the simple method Arm-Pointer, improved method Arm-Pointer, and an ideal case were compared. The working time for detecting an object pointed at is one of the most important indices for user interface evaluation. In the experiment, users pointed at objects indicated randomly and the detecting time was measured. We prepared a laser pointer as the ideal device for pointing, because user can point at precise positions in real time perfectly with a laser pointer, though he can't interact with objects. So we used the measured working time of the laser pointer as the ideal working time value. Working time was evaluated for three types of method: the laser pointer, the simple method, and the improved method.

The number of showing objects for the user was 18 (6 objects x 3 times). Each 3D position of the 6 numbered objects was random on a wall. The sequence of showing objects was random. Four subjects were participated. The object sequence was the same for each. Voxel space resolution was 5 x 5 x 5 for size 2.5 x 2.5 x 2.5 m^3. Voting weights in the improved method are (a, b, c, d) = (10, 5, 2, 1). The object number was indicated automatically on the display in front of the user. The working time from object number indication until pointing finished was measured.

Table 1 shows the results of experiment. Averages of working time were 1.44 [s] for the laser pointer, 3.66 [s] for the simple method, 2.74 [s] for the improved method, respectively. The improved method is faster than the simple method, and the speed is not so slow compared with the laser pointer as an ideal device for pointing. A laser pointer can point at precise positions, but no interactions are available with it and the user needs to keep holding. However, our method has advantages in that it allows interaction with real objects and the user does not have to hold anything. In this experiment, laser pointer has a feedback effect because of projected laser beam. We plan to perform an experiment with feedback for our simple and improved methods and evaluate the methods under more similar conditions in future.

Table 1. Experiment results

Method	Ave.[s]	SD
Laser pointer	1.44	0.33
Simple method	3.66	2.06
Improved method	2.74	1.34

SD: Standard Deviation

7 Conclusion

We proposed a 3D pointing interface called Arm-Pointer that can perform 3D recognition of arm pointing direction. The user does not have to wear any equipment and can point to a target object directly by extending his arm. Since Arm-Pointer can point to real objects in a real space, it realizes an intuitive interface. In the case of either a long-sleeved or short-sleeved shirt, Arm-Pointer can extract the arm pointing direction correctly. Thus, Arm-Pointer will have various applications in the real world. Future work includes considering more efficient voting algorithms especially for the case of a large number of objects, and further improvements on the accuracy of the system.

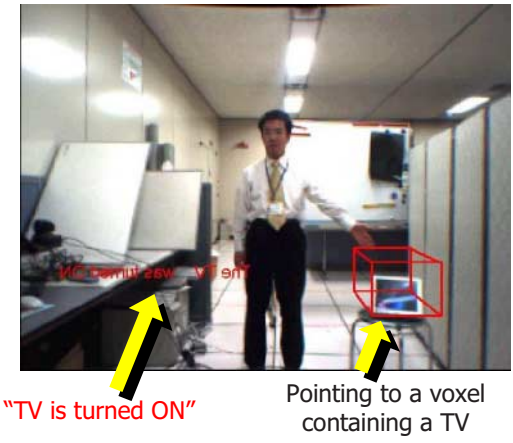


Fig. 9. Interaction with a real object

Acknowledgement

The authors thank Dr. H. Ichino for supporting this research. They also thank the members of the Home Communication Research Group at NTT for helpful discussions.

References

[1] Bolt, R.A. "Put-That-There": Voice and Gesture at the Graphics Interface. Proc. SIGGRAPH 1980, Vol14, No.3, (1980), pp. 262-270.
[2] Tsukada, K. and Yasumura, M. Ubi-Finger: Gesture Input Device for Mobile Use. Proc. APCHI 2002, Vol. 1, (2002), pp. 388-400.
[3] Sugimoto, A., Nakayama, A. and Matsuyama, T. Detecting a Gazing Region by Visual Direction and Stereo Cameras. Proc. ICPR 2002, (2002), pp. 278-282.
[4] Rekimoto, J. SmartSkin: An Infrastructure for Freehand Manipulation on Interactive Surfaces. Proc. CHI 2002, (2002), pp. 113-120.
[5] Koike, H., Sato, Y., Kobayashi, Y., Tobita, H. and Kobayashi, M. Interactive Textbook and Interactive Venn Diagram: Natural and Intuitive Interface on Augmented Desk System. Proc. CHI 2000, (2000), pp. 121-128.
[6] Freeman, W.T. and Weissman, C.D. Television control by hand gestures. Proc. AFGR 1995, (1995), pp. 179-183.
[7] Krueger, M.W., Gionfriddo, T. and Hinrichsen, K. VIDEOPLACE An Artificial Reality. Proc. CHI'85, (1985), pp. 35-40.
[8] Shibuya, Y. and Tamura, H. Interface Using Video Captured Images. Human-Computer Interaction: Ergonomics and User Interfaces, Vol.1, (1999), pp. 247-250.
[9] Morikawa, O., Yamashita, J., Fukui, Y. and Sato, S. Soft initiation in HyperMirror-III. Human-Computer Interaction INTERACT'01, (2001), pp. 415-422.

[10] Hosoya, E., Kitabata, M., Sato, H., Harada, I., Nojima, H., Morisawa, F., Mutoh, S. and Onozawa, A. A Mirror Metaphor Interaction System: Touching Remote Real Objects in an Augmented Reality Environment. Proc. ISMAR'03, (2003), pp. 350-351.
[11] Fukumoto, M., Suenaga, Y. and Mase, K. "FINGER-POINTER": Pointing Interface by Image Processing. Computer & Graphics, Vol. 18, (1994), pp. 633-642.
[12] Wilson, A. and Oliver, N. GWINDOWS: Towards Robust Perception-Based UI. Proc. CVPR 2003, (2003).
[13] Jojic, N., Brumitt, B., Meyers, B., Harris, S. and Huang, T. Detection and Estimation of Pointing Gestures in Dense Disparity Maps. Proc. AFGR 2000, (2000), pp. 468-475.

Hand Gesture Recognition in Camera-Projector System*

Attila Licsár[1] and Tamás Szirányi[1,2]

[1]University of Veszprém, Department of Image Processing and Neurocomputing
H-8200 Veszprém, Egyetem u. 10. Hungary
licsara@almos.vein.hu
[2]Analogical & Neural Computing Laboratory
Computer & Automation Research Institute, Hungarian Academy of Sciences
H-1111 Budapest, Kende u. 13-17, Hungary
sziranyi@sztaki.hu

Abstract. Our paper proposes a vision-based hand gesture recognition system. It is implemented in a camera-projector system to achieve an augmented reality tool. In this configuration the main problem is that the hand surface reflects the projected background, thus we apply a robust hand segmentation method. Hand localizing is based on a background subtraction method, which adapts to the changes of the projected background. Hand poses are described by a method based on modified Fourier descriptors, which involves distance metric for the nearest neighbor classification. The proposed classification method is compared to other feature extraction methods. We also conducted tests on several users. Finally, the recognition efficiency is improved by the recognition probabilities of the consecutive detected gestures by maximum likelihood approach.

1 Introduction

Video projection is widely used for multimedia presentations. In such situations users usually interact with the computer by standard devices (keyboard, mouse). This kind of communication restricts the naturalness of the interaction because the control of the presentation keeps the user in the proximity of the computer. In this paper we demonstrate an effective human-computer interface for a virtual mouse system in a projector-camera configuration. It would be more comfortable and effective if the user could point directly to the display device without any hardware equipment. Our proposed method interacts with the projected presentations or applications by hand gestures in a projector-camera system. For this purpose we use the image acquired by a camera observing the gestures of the speaker in front of the projected image. The system applies a boundary-based method to recognize poses of static hand gestures. The virtual mouse-based application is controlled by the detected hand poses and the palm positions. The virtual user-interface can be displayed onto the projected background image, so the user controls and interacts directly with the projected interface realizing an augmented reality.

* This paper is based on research supported by OTKA-T037829 of the Ministry of Education, Hungary.

N. Sebe et al. (Eds.): HCI/ECCV 2004, LNCS 3058, pp. 83-93, 2004.

In the following we present related systems and give an overview of our work. In the next sections an overview of camera-projector systems and hand segmentation methods are described. Section 4 contains the gesture classification by several feature extraction methods. Finally, we describe an estimation method to increase recognition efficiency by collecting gesture probabilities in time.

2 Related Works

The aim of camera and projector based configurations is that the user interaction should be performed with the projected image instead of applying computer interfaces indirectly. A projector-camera pair is used to display the user interface on the projected surface (Fig. 1) where the camera acquires (camera image) the projected information (projected image) and the gestures of the user provide feedback about the interaction.

Usual methods apply standard white-boards or screens to display the information to the audience. The interaction can be induced by special hardware or vision-based methods. In SmartBoard [0] there are special display hardware devices with sensors e.g. to detect physical contact with the display or use laser pointer for the interaction with user interface by a vision-based system. BrightBoard [0] system uses a video camera and audio feedback to control the computer through painting simple marks onto the board. Other methods, like DigitalDesk [0], FreeHandPresent [0] apply hand gestures e.g. to navigate in a projected presentation by a restricted gesture set by counting and tracking fingers against to the cluttered background. The changing background disturbs the finger finding process so it defines a control area on a white background next to or above the projected surface. The projected image involves restricted background containing only figures and texts. This method applies finger resting on an item for 0.5 seconds as a “pick up” gesture. Magicboard system [0] applies camera and projector pair to get spatio-temporal activities on a white board by finger tracking method. In [0] system works with back-projection screen and can be used for mouse-cursor positioning by pointing with the arm. It determines the arm direction by 3D stereo computation and command is generated by voice signal.

Our goal is the development of a more natural interface in a camera-projector system using hand gesture analysis. The proposed system detects hand poses by shape analysis resulting in a larger vocabulary set in the communication. The correct shape detection is solved in the presence of a cluttered background.

3 System Overview

In our proposed method the arm and the forearm are segmented from the projected background. Our method uses a large gesture vocabulary with 9 hand poses and handles the changes of the complex background. Thus, more tasks can be performed by hand poses in contrast to finger tracking-based methods which result in a restricted gesture set in the interaction.

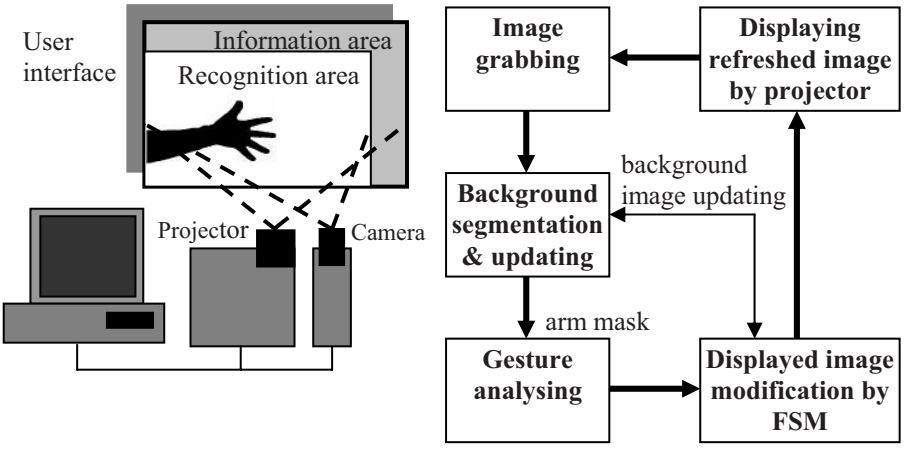


Fig. 1. System configuration and processing scheme

The flow diagram of the proposed method can be seen in the right side of Fig. 1. The camera grabs the projected background images only from a sub-region of the projected surface (recognition area). Out of the view of the camera the projected area can be used to display any information about the state of the recognition process (information area) e.g. pictogram of the detected gesture class. The first step is the foreground segmentation by our background subtraction method. This background image is updated when any change is detected in the projected image. The gesture module analyzes the segmented arm image and the result of the recognition gives the input of a Finite State Machine (FSM). The grammar of the FSM determines the actual task, which modifies the projected image. This task can be a command; e.g. order the system to step the presentation to the next slide or draw a line in the projected image. Because the projected image is known by the system the stored background image can be updated corresponding to the modified projected image. Finally, the projector displays the modified image and the processing cycle starts again. The next background subtraction is accomplished on the updated background image.

3.1 Segmentation Process

In the gesture recognition system the field of view of the camera is a subset of the projected region. Therefore any object in the projector beam reflects the exposure generating different texture patterns on the surface of the arm. For that reason the texture and the color of the hand is continuously changing according to the projected image and object position. These circumstances exclude any color segmentation or region-growing method for the segmentation. In that case the most popular solutions are based on finger tracking [0], but they restrict the usable gesture vocabulary. On that account we chose a background subtraction method and extended it to handle background changing. During projection the reflectance factor of the projected screen is near to 100% while the maximum for human skin is 70%, because the human skin partly absorbs the light, so it behaves as an optical filter [0]. Our method summarizes

image difference with each image channel and foreground objects are classified by this summarized difference image by a threshold value. If the projector ray intensity is small at the position of the hand, e.g. the projected background is black, the difference between the hand and background reflection will be small and noisy. Hence the minimal projector lighting is increased above a threshold intensity value (in our case 20%) by linear histogram transformation of the projected image. The system only transforms the projected image during the interaction if any foreground object is detected by the segmentation method.

Since forearm features do not contain important information, the perfect and consequent segmentation of palm and forearm is important. The problem of automatic segmentation is introduced by other systems [0, 0]. We use a similar width-based wrist locating technique, which uses the main direction of the arm calculated from image moments. Considering this direction of the hand, the width of the wrist and that of the forearm can be measured. Analyzing width parameters of the forearm, the wrist position can be determined using anatomical structure information of the hand, because the calculated width values increase significantly at the wrist points from the forearm to the palm. Result of the segmentation process can be seen on Fig. 2.

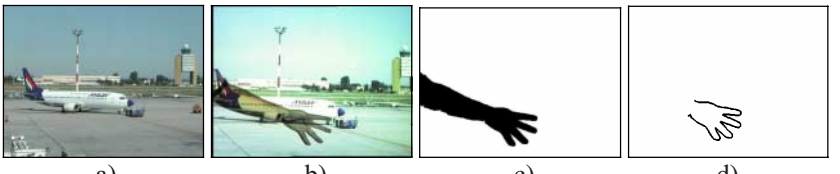


Fig. 2. Steps of the contour segmentation; a) Input image; b) camera image with arm; c) segmented arm image; d) extracted palm contour

3.2 Calibration of the Camera-Projector System

The image grabbed from the camera involves the projected image in the background and any foreground object between the camera and the projected screen. In the camera image these objects are 2D projections of the 3D environment hence the contents of the image suffer from perspective distortions such as keystoning. Consequently, the system needs to register the coordinates of the pixels between the projected and its distorted version, which is grabbed by the camera. In the system this perspective distortion is modelled by a polynomial warping between the coordinates of the camera and the projector images. In our experiments the image warping of a first order polynomial was insufficient because higher order was required for the precise point registration. The second order polynomial equations can be expressed [0] as follows:

$$\begin{aligned} x' &= a_0 + a_1 \cdot x + a_2 \cdot y + a_3 \cdot x^2 + a_4 \cdot xy + a_5 \cdot y^2 \\ y' &= b_0 + b_1 \cdot x + b_2 \cdot y + b_3 \cdot x^2 + b_4 \cdot xy + b_5 \cdot y^2 \end{aligned} \tag{1}$$

where (a_i,b_i) are the weighting coefficients of the geometrical warping, (x,y) the original and (x',y') are the new transformed positions. These input and output sample

points are determined from the projection of a special calibration pattern image, or it can also be done by the edge content of images. Weighting coefficients are chosen to minimize the mean-square error between observed coordinate points and *(x',y')* coordinate points. After the geometrical calibration system may give the correspondence between the original projected and detected palm position.

3.3 Background Image Generation by A Priori Information

The main disadvantage of the background segmentation method is that it fails when background (user interface) significantly changes. In that case the well-known background updating techniques, e.g. running image averaging, does not work because certain regions of the projected image could alter behind the hand. Thus, we improved the segmentation method to overcome this problem by background image generation from the a priori information of the system configuration. However, the input of the projected background image is known, so we could generate an artificial background without any foreground object. In the segmentation process this updated image is used for the background subtraction. The main problems are that the camera-grabbed image suffers from color and geometrical distortion due to perspective projections, and the color transfer function of the camera and the projector. In the color calibration phase an intensity transfer function (look-up table - LUT) is generated from the intensity of the input image and the grabbed image. Sample intensity values are projected and grabbed by the system and these sample pairs (control points) are used for generating the LUT. Values between control points are best interpolated by a fifth order polynomial. The geometrical transformation parameters are determined in the previous section and the image warping uses bilinear interpolation.

When the background changes the system warps the input image by geometrical warping equations, and then it is transformed by the calculated LUT to generate the correct background image for the image differencing. The system generates the background image when it detects any foreground object and the projected image changes significantly. This change detection is performed by a simple image differencing between consecutive projected image frames. Fig. 3. demonstrates segmentation results by generated background image. This segmentation method gives satisfactory results only when used with the color and the geometrically transformed background image. During the interaction this initial background image will be refreshed by the original camera image for precise segmentation by running average method. If any new object appears in the projected image, it is detected by the background segmentation method. If more than one blob is detected, the system chooses the correct one by a labeling method. This labeling method chooses the correct arm object by tracking its last known position and size parameter in the previously segmented images. The labeled arm blob is used for the gesture analyzing. Regions of the residual detected blobs assign regions for the background refreshing. All assigned points of the background image are refreshed with the corresponding pixel from the camera image resulting a continuous background updating.

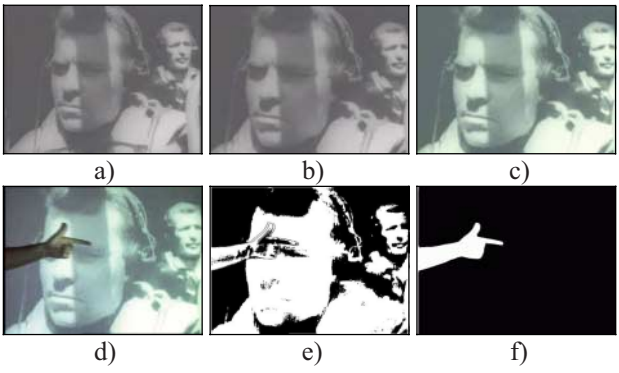


Fig. 3. Background segmentation results; a) the original projected image; b) geometrically transformed projected image; c) geometrical and color transformed projected image; d) real camera image with arm; e) background segmentation of the real camera image with geometrically transformed projected image; f) segmentation result on the geometrical and color transformed image (for color image versions see: http://almos.vein.hu/~licsara/projector)

4 Comparison of Feature Vectors for the Contour Classification

We applied a boundary-based method for the classification. Fourier descriptors are widely used for shape description e.g. character recognition [0], and in content-based image retrieval systems (CBIR) [0]. Recognition methods with Fourier descriptors are usually based on neural networks classification algorithms [0, 0] resulting 90-91% recognition rates for 6 gestures. In our method gesture contour is classified by nearest neighbor rule and the distance metric based on the modified Fourier descriptors [0] (MFD), what is invariant to transition, rotation and scaling of shapes. In these systems the examined shape should be defined by a feature vector, which is a closed curve, so the discrete function is periodic, to expand it into Fourier series. Our new feature vector approach can be seen on Fig. 4A. This feature is a complex coordinate vector (Method "A"), which is generated from the coordinate points of the palm boundary between wrist points as a complex sequence. The problem with that sequence is that it is not periodic, so we need to extend and duplicate it with its reversed sequence to get a periodic function. We compared the previous method with several feature extraction methods (Fig. 4B, C, D), which are widely used in CBIR systems [0]. The second method (Method "B") computes a similar boundary coordinate sequence, but it is generated from the contour of the whole hand mask. The next feature vector is derived from the centroid distance (Method "C") sequence, which is expressed by the distance of the boundary points from the centroid of the silhouette. Finally, the wrist centroid distance (Method "D") feature vector computes distances of the contour points from the centre of the wrist line.

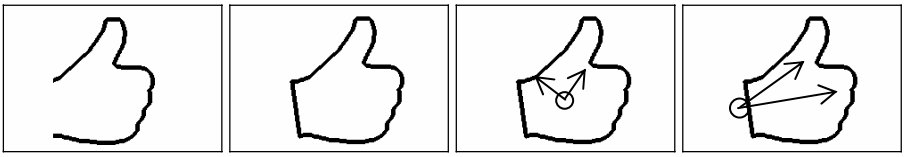


Fig. 4. Examined feature vectors for palm shape classification with the 4 different methods

The extracted feature sequence is classified by the modified Fourier descriptor. The method calculates the discrete Fourier transform (DFT) of this complex sequence. This method applies magnitude values of the DFT coefficients to be invariant to the rotation. We extended the MFD method to get symmetric distance computation. Denoting the DFT coefficients of the compared curves with F_n^1 and F_n^2, standard deviation function denoted by σ, the distance metric between two curves is as follows:

$$Dist(F_n^1, F_n^2) = \sigma\left(\frac{|F_n^1|}{|F_n^2|}\right) + \sigma\left(\frac{|F_n^2|}{|F_n^1|}\right) \tag{2}$$

We examined how many Fourier descriptors should be used in the distance computation. We measured the average efficiency of the recognition with several cut-off frequencies and feature extraction methods (Fig. 5). By determining the appropriate cut-off frequency the classification method is robust against noise of irregularities of the shape boundaries. One advantage of this robust method is that the training set is very small. In our system the training phase is very fast because we store the average feature vectors of 20 consecutive gesture samples for each class.

From the experiments we chose the first 6 coefficients (excluding the DC component) for methods “A”, “C” and “D“, and the first 8 coefficients for method “B”. These results are utilized in our gesture recognition tests. Gestures of several users are tested with the proposed feature extraction methods. We have tested the recognition methods with 9 gesture classes and 400 gesture samples per person (Table 1). Each user trained all gesture classes before the recognition phase. Users can be found in the rows, while different feature extraction methods are in the columns.

It can be seen from our tests that the classification method gives better result, if the feature vector is calculated from the complex sequence of the boundary between wrist points (Method “A”). This approach gives more unambiguous features, since for example the shape contours of the palm when showing only the index or the thumb finger is very similar to each other, while the contour between wrist points are still distinct. This assumption is proved by the experimental results. The proposed system runs in simultaneous real-time performing image projection and grabbing tasks at resolution of 384*288 pixels on a single 1.7GHz Pentium processor.

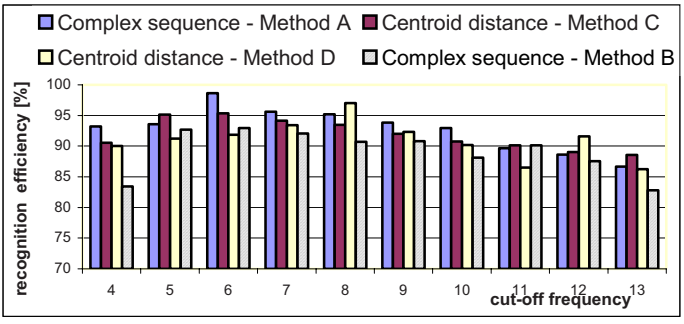


Fig. 5. Recognition efficiency by several cut-off frequencies

Table 1. Pose classification results with several method and users

	Recognition rates [%]			
	Method A	Method B	Method C	Method D
Users				
User 1	99.6	96.8	98.1	97
User 2	98.3	92.6	96.7	98.1
User 3	97.9	95.2	95.3	91.2
User 4	98.2	91.2	94.5	92.3

5 Correction of the Recognition Efficiency

From our experiments we observed that during the interaction users perform gestures for a minimal time period (1-2 sec.). Therefore it can be supposed that results of the recognition should be stable for a given time except when the user changes the performed class. The distance is measured between the actual gesture and stored gesture classes with several consecutive gestures in time. On Fig. 6 the probability order of the faulty classified gesture classes is described. If the recognition of the actual gesture is false, the correct gesture class is not recognized as the most probable gesture (the measured distance corresponds to the probability of the recognized gesture).

It can be seen that most of the misrecognized gestures were classified into the second most probable gesture class with a high probability. Consequently, we could derive gesture probabilities from several consecutive frames, and choose the most

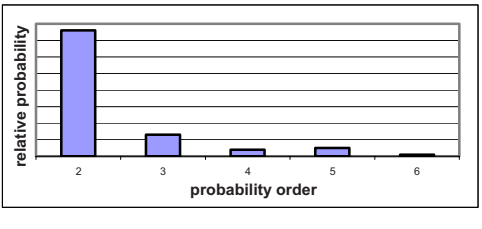


Fig. 6. Measured probability order of the unrecognized gesture classes

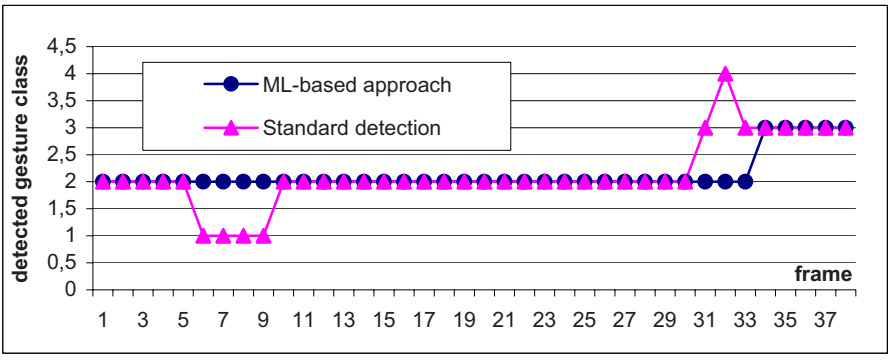


Fig. 7. Improving of the recognition efficiency by analyzing gestures in time

probable gesture class with maximum likelihood. If the occurrence of the unrecognized gestures is low then the detection of the misrecognized gestures can be improved. The estimated gesture class is detected by a maximum likelihood approach:

$$L = \arg\max_{i} \left(\prod_{t} \left(1 - d_{i,t}\right) \right) \tag{3}$$

Parameter $d_{i,t}$ is the measured distance between two gesture classes normalized by the maximum distance value into [0,1[interval. The zero distance means that the probability of the actual class is 1. Probability of the gesture recognition is calculated from the proposed distance metric, where parameter i identifies the gesture class, while t is the time parameter between consecutive frames. The estimated gesture (L) is determined by the maximum likelihood of gesture classes in the time domain. The standard gesture recognition (Section 4) is compared with the above maximum likelihood approach (

Fig. 7). The test user performs 2 gestures with pose changing at frame #31. The recognition with standard method fails from frame #6 to frame #9. The proposed maximum likelihood (ML) estimation corrects this error in time. Between frames #31 and #33 the gesture recognition is unstable due to the gesture transition between two poses. The ML based estimation repairs this error and gives stable recognition result in time. In our experiments ML estimation was calculated in the time period of 6 consecutive frames. In Table 2 the recognition efficiency is summarized applying the standard and the maximum likelihood-based (ML) methods. User tests involve consecutive gesture samples from each user. The ML based method corrects the misrecognized gesture classes and follows the gesture changing in time.

6 Conclusions and Future Works

The vision-based gesture recognition system and the camera-projector configuration form a natural way to control multimedia presentations or manipulate directly the

Table 2. Recognition results with standard and Maximum likelihood-based recognition

	Recognition rates [%]			
	User 1	User 2	User 3	User 4
Methods				
Standard method	99.6	98.3	97.9	98.2
ML-based method	100	99.3	99	99.6

projected image. Our hand pose recognition system offers more freedom in communication for the speaker if compared to other methods using camera-projector systems. The above work has shown that the modified Fourier-based method is robust even with a small training set, so the modification of gesture vocabulary or retraining of gestures is more efficient. We have tested our feature extraction method against several methods from the literature and showed that our method is significantly efficient. We have shown that measuring the detection results by maximum likelihood approach in the time domain could significantly improved the recognition efficiency. Gesture-based systems are considered as typical user-independent recognition tools. Users would like to use them with high recognition efficiency without preliminary training of gestures. In our consecutive work we deal with interactive training of gestures to avoid retraining of all gestures if an untrained user would like to use the system.

References

[1] Streitz, N.A., Geissler J., Haake J.M., Hol, J.: DOLPHIN: Integrated Meeting Support across Liveboards, Local and Remote Desktop Environments. Proceeding of the ACM CSCW, (1994) 345-358

[2] Fraser, Q.S., Robinson, P.: BrightBoard: A Video-Augmented Environment. Proceedings of CHI'96. Vancouver (1996) 134-141

[3] Wellner, P.: Interacting with Paper on the DigitalDesk. Communications of the ACM. (1993)

[4] Hardenberg, C., Berard, F.: Bare-Hand Human Computer Interaction. Proc. of ACM PUI, Orlando (2001)

[5] Hall, D., Gal, C., Martin, J., Chomat, O., Crowley, J.L.: MagicBoard: A contribution to an intelligent office environment. Robotics and Autonomous Systems 35. (2001) 211–220

[6] Leubner, C., Brockmann, C., Müller, H.: Computer-vision-based Human Computer Interaction with a Back Projection Wall Using Arm Gestures. 27th Euromicro Conference. (2001)

[7] Pratt, W.K.: Digital Image Processing, Wiley-Interscience, New York (2001)

[8] Störring, M., Andersen, H. J., Granum, E.: Skin colour detection under changing lighting conditions. 7th Symposium on Intelligent Robotics Systems. Coimbra Portugal 20-23

[9] Imagawa, K., Taniguchi, R., Arita, D., Matsuo, H., Lu, S., Igi, S.: Appearance-based Recognition of Hand Shapes for Sign Language in Low Resolution Image. Proceeding of 4th ACCV. (2000) 943-948

[10] Koh, E.S.: Pose Recognition System. BE Thesis. National University of Singapore (1996)

[11] Rui, Y., She, A., Huang, T.S.: A Modified Fourier Descriptor for Shape Matching in MARS. Image Databases and Multimedia Search. (1998) 165-180
[12] Ng, C.W., Ranganath, S.:Real-time gesture recognition system and application. Image and Vision Computing 20, (2002) 993-1007
[13] Chen, F.S., Fu, C.M., Huang, C.L.: Hand Gesture Recognition Using a Real-Time Tracking Method and Hidden Markov Models. Image and Vision Computing 21, (2003) 745-758
[14] Zhang, D., Lu, G.: A Comparative Study of Fourier Descriptors for Shape representation and Retrieval. ACCV2002. Melbourne Australia (2002)

Authentic Emotion Detection in Real-Time Video

Yafei Sun[1], Nicu Sebe[2], Michael S. Lew[3], and Theo Gevers[2]

[1] School of Computer Science and Engineering, Sichuan University, China
[2] Faculty of Science, University of Amsterdam, The Netherlands
[3] LIACS Media Lab, Leiden University, The Netherlands

Abstract. There is a growing trend toward emotional intelligence in human-computer interaction paradigms. In order to react appropriately to a human, the computer would need to have some perception of the emotional state of the human. We assert that the most informative channel for machine perception of emotions is through facial expressions in video. One current difficulty in evaluating automatic emotion detection is that there are currently no international databases which are based on authentic emotions. The current facial expression databases contain facial expressions which are not naturally linked to the emotional state of the test subject. Our contributions in this work are twofold: First, we create the first authentic facial expression database where the test subjects are showing the natural facial expressions based upon their emotional state. Second, we evaluate the several promising machine learning algorithms for emotion detection which include techniques such as Bayesian Networks, SVMs, and Decision trees.

1 Introduction

In recent years there has been a growing interest in improving all aspects of the interaction between humans and computers. It is argued that to truly achieve effective human-computer intelligent interaction (HCII), there is a need for the computer to be able to interact naturally with the user, similar to the way human-human interaction takes place. Humans interact with each other mainly through speech, but also through body gestures, to emphasize a certain part of the speech and display of emotions. Emotions are displayed by visual, vocal, and other physiological means. There is a growing amount of evidence showing that emotional skills are part of what is called "intelligence" [27, 16]. One of the important way humans display emotions is through facial expressions.

Evaluation of machine learning algorithms generally requires carefully designed ground truth. In facial expression analysis, several test sets exist such as the Cohn-Kanade [18] and JAFFE [21] databases. However, these test sets do not represent the authentic facial expressions for the corresponding emotional state. In these test sets, the subject is asked to mimic the facial expression which may correspond to an emotional state. The subject is not asked to show the natural facial expression corresponding to how he is feeling. Even within these test sets,

N. Sebe et al. (Eds.): HCI/ECCV 2004, LNCS 3058, pp. 94–104, 2004.

the authors (i.e. Kanade et al. [18]) have commented that posed facial behavior is mediated by separate motor pathways than spontaneous facial behavior. As far as we are aware, this is the first attempt to create an authentic emotion database. We shall come back to this subject in Section 2.

While the authentic facial expression test set is important for evaluation and comparison, our fundamental goal is to perform real-time emotion classification using automatic machine learning algorithms. Our real-time system uses a model based non-rigid face tracking algorithm to extract motion features that serve as input to a classifier used for recognizing the different facial expressions and is discussed briefly in Section 3. We were also interested in testing different classifiers from the machine learning literature that can be used for facial expression analysis. We present an extensive evaluation of 24 classifiers using our authentic emotion database (Section 4). We have concluding remarks in Section 5.

2 Authentic Expression Analysis

In many applications of human computer interaction, it is important to be able to detect the emotional state of the person in a natural situation. However, as any photographer can attest, getting a real smile can be challenging. Asking someone to smile often does not create the same picture as an authentic smile. The fundamental reason of course is that the subject often does not feel happy so his smile is artificial and in many subtle ways quite different than a genuine smile.

2.1 Authentic Expression Database

Our goal for the authentic expression database was to create ground truth where the facial expressions would correspond to the current emotional state of the subject. We consulted several members of the psychology department who recommended that the test be constrained as follows to minimize bias. First, the subjects could not know that they were being tested for their emotional state. Knowing that one is in a scientific test can invalidate or bias the results by influencing the emotional state. Second, we would need to interview each subject after the test to find out their true emotional state for each expression. Third, we were warned that even having a researcher in the same room with the subject could bias the results.

We decided to create a video kiosk with a hidden camera which would display segments from recent movie trailers. This method had the main advantages that it would naturally attract people to watch it and we could potentially elicit emotions through different genres of video footage - i.e. horror films for shock, comedy for joy, etc. From over 60 people who used the video kiosk, we were able to get the agreement of 28 students within the computer science department for the database. After each subject had seen the video trailers, they were interviewed to find out their emotional state corresponding to the hidden camera video footage. We also secured agreement for the motion data from their video

footage to be distributed to the scientific community which is one of the primary goals for this database.

In this kind of experiment, we can only capture the expressions corresponding to the naturally occurring emotions. This means that our range of emotions for the database was constrained to the ones genuinely felt by the subjects. For this database, the emotions found were either (1) Neutral; (2) Joy; (3) Surprise, or (4) Disgust. From having created the database, some items of note based purely on our experiences: (1) It is very difficult to get a wide range of emotions for all of the subjects. Having all of the subjects experience genuine sadness for example is difficult. (2) The facial expressions corresponding to the internal emotions is often misleading. Some of the subjects appeared to be sad when they were actually happy. (3) Students are usually open to having the data extracted from the video used for test sets. The older faculty members were generally not agreeable to being part of the database.

2.2 Posed versus Authentic Expressions

In selecting facial stimuli, the issue of whether to use posed or spontaneous expressions has been hotly debated. Experimentalists and most emotion theorists argue that spontaneous expressions are the only "true" expressions of facial emotion and therefore such stimuli are the only ones of merit.

When recording authentic facial expressions several aspects should be considered. Not all people express emotion equally well; many individuals have idiosyncratic methods of expressing emotion as a result of personal, familial, or culturally learned display rules. Situations in which authentic facial expression are usually recorded (e.g., laboratory) are often unusual and artificial. If the subject is aware of being photographed or filmed, facial expressions may not be spontaneous anymore. Even if the subject is unaware of being filmed, the laboratory situation may not encourage natural or usual emotion response. In interacting with scientists or other authorities, subjects will attempt to act in appropriate ways so that emotion expression may be masked or controlled. Additionally, there are only a few universal emotions and only some of these can be ethically stimulated in the laboratory.

On the other hand, posed expressions may be regarded as an alternative, provided that certain safeguards are followed. Increased knowledge about the face, based in large part on observation of spontaneous, naturally occurring facial expressions, has made possible a number of methods of measuring the face. These measurement techniques can be used to ascertain whether or not emotional facial behavior has occurred and what emotion is shown in a given instance. Such facial scoring provides a kind of stimulus criterion validity that is important in this area. Additionally, posers can be instructed, not to act or pose a specific emotion, but rather to move certain muscles so as to effect the desired emotional expression. In this way, experimental control may be exerted on the stimuli and the relationship between the elements of the facial expression and the responses of observers may be analyzed and used as a guide in item selection.

From the above discussion, it is clear that the authentic facial expression analysis should be performed whenever is possible. Posed expression may be used as an alternative only in restricted cases and they can be mostly used for benchmarking the authentic expressions.

3 Facial Expression Recognition

Extensive studies of human facial expressions performed by Ekman [11, 12] gave evidence to support universality in facial expressions. According to these studies, the "universal facial expressions" are those representing happiness, sadness, anger, fear, surprise, and disgust. To code facial expressions, Ekman and Friesen [12] developed the Facial Action Coding System (FACS) in which the movements on the face are described by a set of action units (AUs) which have some related muscular basis. Ekman's work inspired many researchers to analyze facial expressions by means of image and video processing. By tracking facial features and measuring the amount of facial movement, they attempt to categorize different facial expressions. Recent work on facial expression analysis and recognition [4, 30, 9, 2, 23] has used these "basic expressions" or a subset of them. The two recent surveys in the area [24, 13] provide an in depth review of many of the research done in recent years. All the methods developed are similar in that they first extract some features from the images or video, then these features are used as inputs into a classification system, and the outcome is one of the preselected emotion categories. They differ mainly in the features extracted and in the classifiers used to distinguish between the different emotions.

Our real time facial expression recognition system (described in Section 3.1) is composed of a face tracking algorithm which outputs a vector of motion features of certain regions of the face. The features are used as inputs to one of the classifiers described in Section 3.2.

3.1 Our Real-Time System

A snap shot of our real-time system with the face tracking and the recognition result is shown in Figure 1. The face tracking we use is based on a system developed by Tao and Huang [28] called the Piecewise Bézier Volume Deformation (PBVD) tracker. This face tracker uses a model-based approach where an explicit 3D wireframe model of the face is constructed. In the first frame of the image sequence, landmark facial features such as the eye corners and mouth corners are selected interactively. A generic face model is then warped to fit the selected facial features. The face model consists of 16 surface patches embedded in Bézier volumes. The surface patches defined this way are guaranteed to be continuous and smooth. The shape of the mesh can be changed by changing the locations of the control points in the Bézier volume.

Once the model is constructed and fitted, head motion and local deformations of the facial features such as the eyebrows, eyelids, and mouth can be

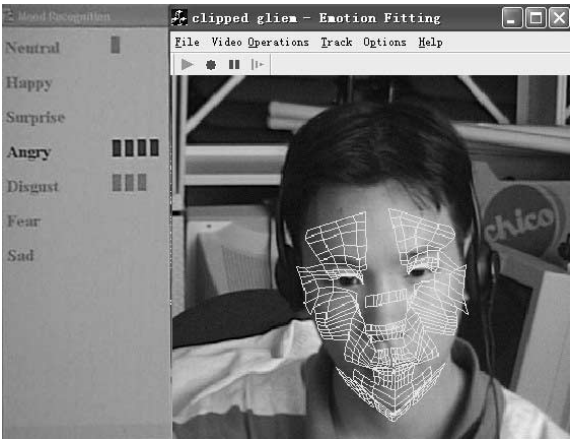
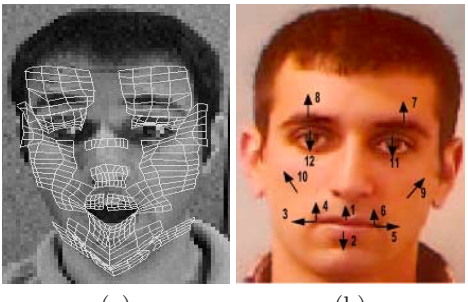


Fig. 1. A snap shot of our realtime facial expression recognition system. On the right side is a wireframe model overlayed on a face being tracked. On the left side the correct expression, Angry, is detected (the bars show the relative probability of Angry compared to the other expressions)



(a) (b)

Fig. 2. (a) The wireframe model, (b) the facial motion measurements

tracked. First the 2D image motions are measured using template matching between frames at different resolutions. Image templates from the previous frame and from the very first frame are both used for more robust tracking. The measured 2D image motions are modeled as projections of the true 3D motions onto the image plane. From the 2D motions of many points on the mesh, the 3D motion can be estimated by solving an overdetermined system of equations of the projective motions in the least squared sense. Figure 2(a) shows an example from one frame of the wireframe model overlayed on a face being tracked.

The recovered motions are represented in terms of magnitudes of some predefined motion of various facial features. Each feature motion corresponds to a simple deformation on the face, defined in terms of the Bézier volume control parameters. We refer to these motions vectors as Motion-Units (MU's). Note

that they are similar but not equivalent to Ekman's AU's and are numeric in nature, representing not only the activation of a facial region, but also the direction and intensity of the motion. The MU's used in the face tracker are shown in Figure 2(b). The MU's are used as the basic features for the classifiers described in the next section.

3.2 Classifiers

Several classifiers from the machine learning literature were considered in our system and are listed below. We give a brief description for each of the classifiers and ask the reader to get more details from the original references. We also investigated the use of voting algorithms to improve the classification results.

Generative Bayesian Networks Classifiers. Bayesian networks can represent joint distributions we usethem to compute the posterior probability of a set of *labels* given the observable *features*, and then we classify the features with the most probable label.

A Bayesian network is composed of a directed acyclic graph in which every node is associated with a variable X_i and with a conditional distribution $p(X_i|\Pi_i)$, where Π_i denotes the parents of X_i in the graph. The directed acyclic graph is the *structure*, and the distributions $p(X_i|\Pi_i)$ represent the *parameters* of the network. A Bayesian network classifier is a *generative* classifier when the class variable is an ancestor (e.g., parent) of some or all features. We consider three examples of generative Bayesian Networks: (1) Naive-Bayes classifier [10] (**NB**) makes the assumption that all features are conditionally independent given the class label. Although this assumption is typically violated in practice, NB have been used successfully in many classification problems. Better results may be achieved by discretizing the continuous input features yielding the **NBd** classifier. (2) The Tree-Augmented Naive-Bayes classifier [15] (**TAN**) attempts to find a structure that captures the dependencies among the input features. In the structure of the TAN classifier, the class variable is the parent of all the features and each feature has at most one other feature as a parent, such that the resultant graph of the features forms a tree. (3) The Stochastic Structure Search classifier [7] (**SSS**) goes beyond the simplifying assumptions of NB and TAN and searches for the correct Bayesian network structure focusing on classification. The idea is to use a strategy that can efficiently search through the whole space of possible structures and to extract the ones that give the best classification results.

The Decision Tree Inducers. The purpose of the decision tree inducers is to create from a given data set an efficient description of a classifier by means of a decision tree. The decision tree represents a data structure which efficiently organizes descriptors. The purpose of the tree is to store an ordered series of descriptors. As one travels through the tree he is asked questions and the answers determine which further questions will be asked. At the end of the path is

a classification. When viewed as a black box the decision tree represents a function of parameters (or descriptors) leading to a certain value of the classifier. We consider the following decision tree algorithms and use their $\mathcal{MLC}$++ implementation [19]: (1) **ID3** is a very basic decision tree algorithm with no pruning based on [25]. (2) **C4.5** is an extension of ID3 that accounts for unavailable values, continuous attribute value ranges, and pruning of decision trees [26]. (3) **MC4** is similar to C4.5 [26] with the exception that unknowns are regarded as a separate value. The algorithm grows the decision tree following the standard methodology of choosing the best attribute according to the evaluation criterion. After the tree is grown, a pruning phase replaces subtrees with leaves using the same pruning algorithm that C4.5 uses. (4) **OC1** is the Oblique decision tree algorithm by Murthy et al [22]. It combines deterministic hill-climbing with two forms of randomization to find a good oblique split (in the form of a hyperplane) at each node of a decision tree.

Other Inducers. (1) Support vector machines [29] **SVM** were developed based on the Structural Risk Minimization principle from statistical learning theory. They are one of the most popular classifiers and can be applied to regression, classification, and density estimation problems. (2) **kNN** is the instance-based learning algorithm (nearest-neighbor) by Aha [1]. This is a good, robust algorithm, but slow when there are many attributes. (3) **PEBLS** is the Parallel Exemplar-Based Learning System by Cost and Salzberg [8]. This is a nearest-neighbor learning system designed for applications where the instances have symbolic feature values. (4) **CN2** is the direct rule induction algorithm by Clark and Niblett [6]. This algorithm inductively learns a set of propositional if...then... rules from a set of training examples. To do this, it performs a general-to-specific beam search through rule-space for the "best" rule, removes training examples covered by that rule, then repeats until no more "good" rules can be found. (5) **Winnow** is the multiplicative algorithm described in [20]. (6) **Perceptron** is the simple algorithm described in [17]. Both Perceptron and Winnow are classifiers that build linear discriminators and they are only capable of handling continuous attributes with no-unknowns and two-class problem. For our multi-class problem we implemented several classifiers, each classifying one class against the rest of the classes and in the end we averaged the results.

Voting Algorithms. Methods for voting classification, such as Bagging and Boosting (AdaBoost) have been shown to be very successful in improving the accuracy of certain classifiers for artificial and real-world datasets [3]. A voting algorithm takes an inducer and a training set as input and runs the inducer multiple times by changing the distribution of training set instances. The generated classifiers are then combined to create a final classifier that is used to classify the test set.

The **bagging** algorithm (**B**ootstrap **agg**regat**ing**) by Breiman [5] votes classifiers generated by different bootstrap samples (replicates). A bootstrap sample is generated by uniformly sampling m instances from the training set with

replacement. T bootstrap samples $B_1, B_2, \ldots, B_T$ are generated and a classifier C_i is built from each bootstrap sample B_i. A final classifier C^* is built from $C_1, C_2, \ldots, C_T$ whose output is the class predicted most often by its sub-classifiers, with ties broken arbitrarily. Bagging works best on unstable inducers (e.g., decision trees), that is, inducers that suffer from high variance because of small perturbations in the data. However, bagging may slightly degrade performance of stable algorithms (e.g. kNN) because effectively smaller training sets are used for training each classifier.

Like bagging **AdaBoost** (**Ada**ptive **Boost**ing) algorithm [14] generates a set of classifiers and votes them. The AdaBoost however, generates classifiers sequentially, while bagging can generate them in parallel. AdaBoost also changes the weights of the training instances provided as input to each inducer based on classifiers that were previously built. The goal is to force the inducer to minimize the expected error over different input distributions. Given an integer T specifying the number of trials, T weighted training sets $S_1, S_2, \ldots, S_T$ are generated in sequence and T classifiers $C_1, C_2, \ldots, C_T$ are built. A final classifier C^* is formed using a weighted voting scheme: the weight of each classifier depends upon its performance on the training set used to build it.

4 Facial Expression Recognition Experiments

In our experiments we use the authentic database described in Section 2. For this database we have a small number of frames for each expression which makes insufficient data to perform person dependent tests. We measure the classification error of each frame, where each frame in the video sequence was manually labeled to one of the expressions (including neutral). This manual labeling can introduce some 'noise' in our classification because the boundary between Neutral and the expression of a sequence is not necessarily optimal, and frames near this boundary might cause confusion between the expression and the Neutral. A different labeling scheme is to label only some of the frames that are around the peak of the expression leaving many frames in between unlabeled. We did not take this approach because a real-time classification system would not have this information available to it.

When performing the error estimation we used n-fold cross-validation (n=10 in our experiments) in which the dataset was randomly split into n mutually exclusive subsets (the folds) of approximately equal size. The inducer is trained and tested n times; each time tested on a fold and trained on the dataset minus the fold. The cross-validation estimate of error is the average of the estimated errors from the n folds. To show the statistical significance of our results we also present the 95% confidence intervals for the classification errors.

We show the results for all the classifiers in Table 1. Surprisingly, the best classification results are obtained with the kNN classifier (k=3 in our experiments). This classifier is a distance-based classifiers and does not assume any model. It seems that facial expression recognition is not a simple classification problem and all the models tried (e.g., NB, TAN, or SSS) were not able to

Table 1. Classification errors for facial expression recognition together with their 95% confidence intervals

Classifiers	Classification Error	Classifiers	Classification Error
NB	8.46 ± 0.93%	MC4	8.45 ± 0.94%
NB bagging	8.35 ± 0.92%	MC4 bagging	7.35 ± 0.76%
NB boosting	8.25 ± 0.97%	MC4 boosting	5.84 ± 0.78%
NBd	8.46 ± 0.93%	OC1	9.05 ± 1.10%
NBd bagging	9.26 ± 1.15%	SVM	13.23 ± 0.93%
NBd boosting	8.65 ± 1.03%	kNN	4.43 ± 0.97%
TAN	6.46 ± 0.34%	kNN bagging	4.53 ± 0.97%
SSS	5.89 ± 0.67%	kNN boosting	4.43 ± 0.97%
ID3	9.76 ± 1.00%	PEBLS	6.05 ± 1.09%
ID3 bagging	7.45 ± 0.66%	CN2	9.26 ± 0.82%
ID3 boosting	6.96 ± 1.00%	Winnow	12.07 ± 1.87%
C4.5	8.45 ± 0.91%	Perceptron	7.75 ± 1.41%

entirely capture the complex decision boundary that separates the different expressions. This argumentation may also explain the surprisingly poor behavior of the SVM.

kNN may give the best classification results but it has its own disadvantages: it is computationally slow and needs to keep all the instances in the memory. The main advantage of the model-based classifiers is their ability to incorporate unlabel data [7]. This is very important since labeling data for emotion recognition is very expensive and requires expertise, time, and training of subjects. However, collecting unlabel data is not as difficult. Therefore, it is beneficial to be able to use classifiers that are learnt with a combination of some labeled data and a large amount of unlabeled data. Another important aspect is that the voting algorithms improve the classification results of the decision trees algorithms but do not significantly improve the results of the more stable algorithms such as NB and kNN.

We were also interested to investigate how the classification error behaves when more and more training instances are available. The corresponding learning curves are presented in Figure 3. As expected kNN improves significantly as more data are used for training.

5 Summary and Discussion

In this work we presented our efforts in creating an authentic facial expression database based on spontaneous emotions. We created a video kiosk with a hidden camera which displayed segments of movies and was filming several subjects that showed spontaneous emotions. One of our main contribution in this work was to create a database in which the facial expressions correspond to the true emotional state of the subjects. As far as we are aware this is the first attempt to

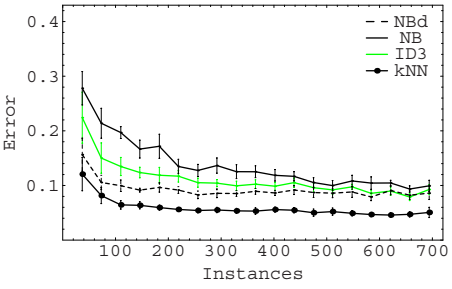


Fig. 3. The learning curve for different classifiers. The vertical bars represent the 95% confidence intervals

create such a database and our intention is to make it available to the scientific community.

Furthermore, we tested and compared a wide range of classifiers from the machine learning community including Bayesian Networks, decision trees, SVM, kNN, etc. We also considered the use of voting classification schemes such as bagging and boosing to improve the classification results of the classifiers. We demonstrated the classifiers for facial expression recognition using our authentic database. Finally, we integrated the classifiers and a face tracking system to build a real time facial expression recognition system.

References

[1] D. W. Aha. Tolerating noisy, irrelevant and novel attributes in instance-based learning algorithms. *International Journal of Man-Machine Studies*, 36(1):267–287, 1992. 100

[2] M. S. Bartlett, I. Littlewort, G.and Fasel, and J. R. Movellan. Real time face detection and expression recognition: Development and application to human-computer interaction. In *CVPR Workshop on Computer Vision and Pattern Recognition for Human-Computer Interaction*, 2003. 97

[3] E. Bauer and R. Kohavi. An empirical comparison of voting classification algorithms:Bagging, boosting, and variants. *Machine Learning*, 36:105–142, 1999. 100

[4] F. Bourel, C. Chibelushi, and A. Low. Robust facial expression recognition using a state-based model of spatially-localised facial dynamic. In *Int. Conference on Automatic Face and Gesture Recognition*, pages 113–118, 2002. 97

[5] L. Breiman. Bagging predictors. *Machine Learning*, 24:123–140, 1996. 100

[6] P. Clark and T. Niblett. The CN2 induction algorithm. *Machine Learning*, 3(4):261–283, 1989. 100

[7] I. Cohen, N. Sebe, F. G. Cozman, and T. S. Huang. Semi-supervised learning for facial expression recognition. In *ACM Workshop on Multimedia Information Retrieval*, pages 17–22, 2003. 99, 102

[8] S. Cost and S. Salzberg. A weighted nearest neighbor algorithm for learning with symbolic features. *Machine Learning*, 10(1):57–78, 1993. 100

[9] G. Donato, M. S. Bartlett, J. C. Hager, P. Ekman, and T. J. Sejnowski. Classifying facial actions. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21(10):974–989, 1999. 97
[10] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. John Wiley and Sons, 1973. 99
[11] P. Ekman. Strong evidence for universals in facial expressions: A reply to Russell's mistaken critique. *Psychological Bulletin*, 115(2):268–287, 1994. 97
[12] P. Ekman and W. V. Friesen. *Facial Action Coding System: Investigator's Guide*. Consulting Psychologists Press, 1978. 97
[13] B. Fasel and J. Luettin. Automatic facial expression analysis: A survey. *Pattern Recognition*, 36:259–275, 2003. 97
[14] Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In *International Conference on Machine Learning*, pages 148–156, 1996. 101
[15] N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29(2):131–163, 1997. 99
[16] D. Goleman. *Emotional Intelligence*. Bantam Books, New York, 1995. 94
[17] J. Hertz, A. Krogh, and R. G. Palmer. *Introduction to the Theory of Neural Computation*. Addison Wesley, 1991. 100
[18] T. Kanade, J. Cohn, and Y. Tian. Comprehensive database for facial expression analysis. In *Int. Conference on Automatic Face and Gesture Recognition*, pages 46–53, 2000. 94, 95
[19] Ron Kohavi, Dan Sommerfield, and James Dougherty. Data mining using $\mathcal{MLC}$++: A machine learning library in C++. *International Journal on Artificial Intelligence Tools*, 6(4):537–566, 1997. 100
[20] N. Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 10(1):57–78, 1993. 100
[21] M. Lyons, A. Akamatsu, M. Kamachi, and J. Gyoba. Coding facial expressions with Gabor wavelets. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 200–205, 1998. 94
[22] S. K. Murthy, S. Kasif, and S. Salzberg. A system for the induction of oblique decision trees. *Journal of Artificial Intelligence Research*, 2:1–33, 1994. 100
[23] N. Oliver, A. Pentland, and F. Bérard. LAFTER: A real-time face and lips tracker with facial expression recognition. *Pattern Recognition*, 33:1369–1382, 2000. 97
[24] M. Pantic and L. J. M. Rothkrantz. Automatic analysis of facial expressions: The state of the art. *IEEE Trans. on PAMI*, 22(12):1424–1445, 2000. 97
[25] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986. 100
[26] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufman, 1993. 100
[27] P. Salovey and J. D. Mayer. Emotional intelligence. *Imagination, Cognition, and Personality*, 9(3):185–211, 1990. 94
[28] H. Tao and T. S. Huang. Connected vibrations: A modal analysis approach to non-rigid motion tracking. In *CVPR*, pages 735–740, 1998. 97
[29] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995. 100
[30] Y. Zhang and Q. Ji. Facial expression understanding in image sequences using dynamic and active visual information fusion. In *ICCV*, pages 113–118, 2003. 97

Hand Pose Estimation Using Hierarchical Detection

B. Stenger[1⋆], A. Thayananthan[1], P.H.S. Torr[2], and R. Cipolla[1]

[1] University of Cambridge, Department of Engineering
Cambridge, CB2 1PZ, UK
{bdrs2,at315,cipolla}@eng.cam.ac.uk
[2] Oxford Brookes University, Department of Computing
Oxford OX33 1HX, UK
philiptorr@brookes.ac.uk

Abstract. This paper presents an analysis of the design of classifiers for use in a hierarchical object recognition approach. In this approach, a cascade of classifiers is arranged in a tree in order to recognize multiple object classes. We are interested in the problem of recognizing multiple patterns as it is closely related to the problem of locating an articulated object. Each different pattern class corresponds to the hand in a different pose, or set of poses. For this problem obtaining labelled training data of the hand in a given pose can be problematic. Given a parametric 3D model, generating training data in the form of example images is cheap, and we demonstate that it can be used to design classifiers almost as good as those trained using non-synthetic data. We compare a variety of different template-based classifiers and discuss their merits.

1 Introduction

This paper considers the problem of locating and tracking an articulated object using a single camera. The method is illustrated by the problem of hand detection and tracking. There is a lot of ambiguity in the problem of tracking a complex articulated object, and a successful method should be able to maintain multi-modal distributions over time. A number of different techniques have been suggested to deal with multi-modality, e.g. particle filtering [7, 11]. When track is lost, a robust tracker should devise a recovery strategy, as for example in region based tracking [23]. This task, however, can be seen as a detection problem, and thus in [19, 20] it is argued that the tracking of complex objects should involve the close synthesis of object detection and tracking.

Object recognition is typically considered as the task of detecting a single class of objects $\mathcal{O}$ (e.g. faces) in a scene $\mathcal{I}$; locating the object in the scene and determining its pose. But suppose we are interested in recognizing m categories of objects $\mathcal{O}_1, \ldots, \mathcal{O}_m$ simultaneously, i.e. are any of a set of objects in the scene, and if so where? How can it efficiently be decided whether the scene contains

⋆ The author is currently with Toshiba Research, Kawasaki, Japan.

N. Sebe et al. (Eds.): HCI/ECCV 2004, LNCS 3058, pp. 105–116, 2004.

one of these objects? One option that is commonly followed is to independently train a classifier for each object [16]. The drawback of such an approach is that computation time scales roughly linearly in the number of objects to be identified. Recently advances have been made in face detection based on the idea of a cascade of classifiers [15, 22], where successively more complex classifiers are combined in a cascade structure, which increases the speed of the detector by focusing attention on promising regions of the image, see figure 1(a). First the image is divided into a set of subregions. The initial classifier eliminates a large portion of these subwindows with little computation; those remaining are processed further down the cascade. At each level the number of subwindows remaining decreases, allowing for more computationally expensive classifiers to be used at the bottom level for accurate discrimination of the remaining subwindows. As the motivation for such a cascade is the minimization of computation time, this paper examines some of the issues involved in classifier design for efficient template-based classification for hand pose estimation.

The next section reviews related work on hierarchical detection and describes the links to 3D pose estimation. Section 3 introduces the shape and colour features that are used, as well as the templates for classification. An evaluation of these classifiers in terms of performance and efficiency is presented in section 4, and their application to a pose estimation problem is shown.

2 Pose Estimation Using Shape Templates

One question is whether a cascaded approach can be used for recognizing multiple objects, i.e. how to design a computationally efficient cascade of classifiers for a given set of objects, $\mathcal{O}_1, \ldots, \mathcal{O}_m$. The problem is closely linked to the recognition of articulated objects which can be thought of as an infinite collection of objects indexed by the joint articulation parameters. In particular, Gavrila [6] examines the problem of detecting pedestrians. Chamfer matching [2] is used to detect humans in different poses, and detecting people is formulated as a template matching problem. When matching many similar templates to an image, a significant speed-up can be achieved by forming a template hierarchy and using a coarse to fine search [6, 14]. Toyama and Blake [21] use exemplar templates to evaluate likelihoods within a probabilistic tracking framework. Shape templates of a walking person are clustered and only the chamfer cost of the prototypes needs to be computed. However, with increasing object complexity the number of exemplars required for tracking rises as well. If a parametric 3D object model is available, the generation of training examples is cheap. Additionally, each generated 2D template is annotated with the 3D model parameters, thus pose recovery can be formulated as object detection: create a database of model-generated images and use a nearest-neighbour search to find the best match. This approach is followed, for example, by Athitsos and Sclaroff for hand pose estimation [1] and Shakhnarovich *et al.* [17] for upper body pose estimation. In [19] it is suggested to partition the parameter space of a 3D hand model using a multi-resolution grid. A distribution is defined on the finest grid and is propagated over time.

Algorithm 1 : Cascade of Classifiers for Multiple Categories

for each subwindow **I**
  start at root node $(l, k) = (1, 1)$.
  if $C_k^l(\mathbf{I}) > 0$ **then** repeat for child nodes of (l, k).
  else assign zero probability to all child nodes of (l, k).
  endif
endfor

This has the advantage that temporal information can be used to resolve ambiguous situations and to smooth the motion. Shape templates, generated by the 3D model, are used to evaluate the likelihoods in regions of the state space. The templates are arranged in a hierarchy and are used to rapidly discard regions with low probability mass. For the first frame, the tree corresponds to a detection tree, thus the idea of cascaded classifiers can be applied, which eliminate large regions of the parameter space at early stages and focus computation on ambiguous regions. In terms of classifiers, the aim is to maintain a high detection rate while rejecting as many false positives as possible at each node in the tree. Within this paper we will analyze the design of cascaded classifiers for such a hierarchy, which can be used within the tree-based filtering framework of [19].

Given a tree which at each level partitions the set of models into mutually exclusive regions $\mathcal{S}_i^l$ for $l = 1, \ldots, L$ where L is the number of levels in the tree and $i = 1 \ldots N_L$ where N_L is the number of sets at that level. So that $\mathcal{S} = \cup_i \mathcal{S}_i^l$ and $\mathcal{S}_j^l \cap \mathcal{S}_k^l = 0; \forall j, k$. The goal is to design a classifier $\mathcal{C}_j^l$ which achieves high detection rates with modest false positive rates or the region $\mathcal{S}_j^l$. The search then proceeds as shown in algorithm 1. A schematic of this algorithm is shown in figure 1b. The next section examines a number of different classifiers based on edge and colour features.

3 Explanation of Features and Classifiers

Edges (occluding contours) and colour (silhouette interior) have proved useful features for recognizing hands and discriminating between different poses, e.g. [1, 13]. Each feature is treated independently, assuming that in different settings one of the features may be sufficient for recognition. Edge features are considered in the following section.

3.1 Edge Features

When using edges as features, robust similarity functions need to be used when comparing a template with the image, i.e. ones that are tolerant to small shape changes. One way to achieve this is to blur the edge image or template before correlating them. Other methods, which are tolerant to small shape deformations and some occlusion are the (truncated) chamfer and Hausdorff distance functions [2, 10]. Both methods are made efficient by the use of fast operations like

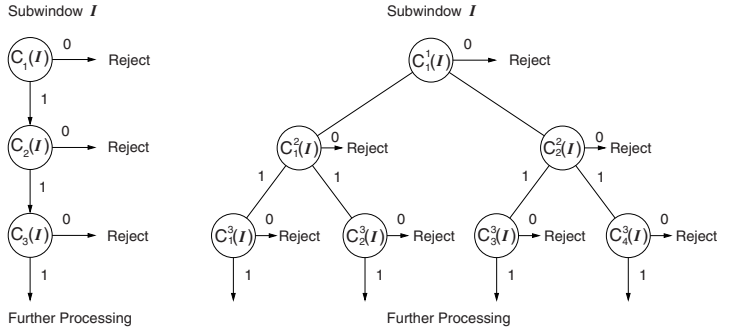


Fig. 1. Cascade of Classifiers (a) A cascade of classifiers for a single object class where each classifier has a high detection and moderate false positive rate **(b)** Classifiers in a tree structure; in a tree-based object recognition scheme each leaf corresponds to a single object class. When objects in the subtrees have similar appearance, classifiers can be used to quickly prune the search. A binary tree is shown here, but the branching factor can be larger than two

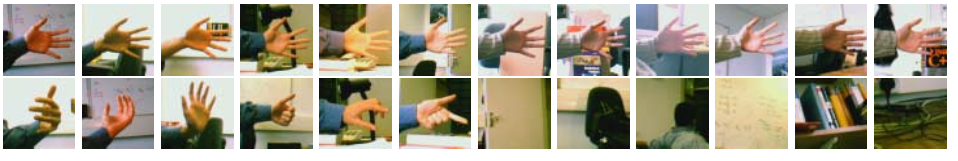


Fig. 2. Examples of Training and Test Images. Top: First six images from left: training images, last six: test images **Bottom:** First six images from left: negative training examples containing hand in different pose, last six: negative examples containing background

the distance transform or dilation of the edge image. Olson and Huttenlocher [14] include edge orientation in Hausdorff matching. This is done by decomposing both template and edge image into a number of separate channels according to edge orientation. The distance is computed separately for each channel, and the sum of these is the total cost.

Both chamfer and Hausdorff matching can be viewed as special cases of linear classifiers [5]. Let $\mathbf{B}$ be the feature map of an image region, for example a binary edge image. The template $\mathbf{A}$ is of the same size and can be thought of as a prototype shape. The problem of recognition is to decide whether or not $\mathbf{B}$ is an instance of $\mathbf{A}$. By writing the entries of matrices $\mathbf{A}$ and $\mathbf{B}$ into vectors $\mathbf{a}$ and $\mathbf{b}$, respectively, this problem can be written as a linear classification problem with a discriminant function $\mathbf{a}^{\mathrm{T}}\mathbf{b} = c$, with constant c. This generalization also permits negative coefficients of $\mathbf{a}$, potentially increasing the cost of cluttered image areas, and different weights may be given to different parts of the shape. Felzenszwalb [5] has shown that a single template $\mathbf{A}$ and a dilated edge map $\mathbf{B}$ is sufficient to detect a variety of shapes of a walking person. A classifier is thus

defined by the entries in the matrix $\mathbf{A}$ and in this paper the following classifiers are evaluated (illustrated in figure 3 (a)). For each type two sets of templates are generated, one with and one without orientation information. For oriented edges the angle space is subdivided into six discrete intervals, resulting in a template for each orientation channel.

Centre Template: This classifier uses a single shape template $\mathbf{A}$, generated using the centre of a region in parameter space. Two possibilities for the feature matrix $\mathbf{B}$ are compared. One is the distance transformed edge image in order to compute the truncated chamfer distance [6]. For comparison, the Hausdorff fraction is computed using the dilated edge image [9]. The parameters for both methods are set by testing the classification performance on a test set of 5000 images. Values for the chamfer threshold τ from 2 to 120 were tested, and $\tau = 50$ was chosen, but little variation was observed for values larger than 20. For the dilation parameter δ values from 1 to 11 were compared, and $\delta = 3$ showed the best performance.

Marginalized Template: In order to construct a classifer which is sensitive to a particular region in parameter space, the template $\mathbf{A}$ is constructed by densely sampling the values in this region, and simultaneously setting the model parameters to these values. The resulting model projections are then pixel-wise added and smoothed. Different versions of matrices $\mathbf{A}$ are compared: (a) the pixel-wise average of model projections, (b) the pixel-wise average, additionally setting the background weights uniformly to a negative value such that the sum of coefficients is zero, and (c) the union of all projections, resulting in a binary template.

Linear Classifier Learnt from Image Data: The template $\mathbf{A}$ is obtained by learning a classifier as described by Felzenszwalb [5]. A labelled training set containing 1,000 positive examples and 4,000 negative examples of which 1,000 contain the hand in a different pose and 3,000 images contain background regions (see figure 2) is used to train a linear classifier by minimizing the perceptron cost function [4].

3.2 Colour Features

Given an input image region, define the feature matrix $\mathbf{B}^s$ as the log-likelihood map of skin colour and $\mathbf{B}^{bg}$ as the log-likelihood map of background colour. Skin colour is represented as a Gaussian in (r, g)-space, and the background distribution is modelled as a uniform distribution. A silhouette template $\mathbf{A}$ is defined as containing $+1$ at locations within the hand silhouette and zero otherwise. Writing these matrices as vectors, a cost function, which corresponds to the log-likelihood [18] can be written as:

$$\mathbf{a}^{\mathrm{T}}(\mathbf{b}^s - \mathbf{b}^{bg}) + \mathbf{1}^{\mathrm{T}}\mathbf{b}^{bg} \quad , \tag{1}$$

where the entries in $\mathbf{b}^{bg}$ are constant when using a uniform background model. Thus, correlating the matrix $\mathbf{B} = \mathbf{B}^s - \mathbf{B}^{bg}$ with a silhouette template $\mathbf{A}$ corresponds to evaluating the log-likelihood of an input region up to an additive

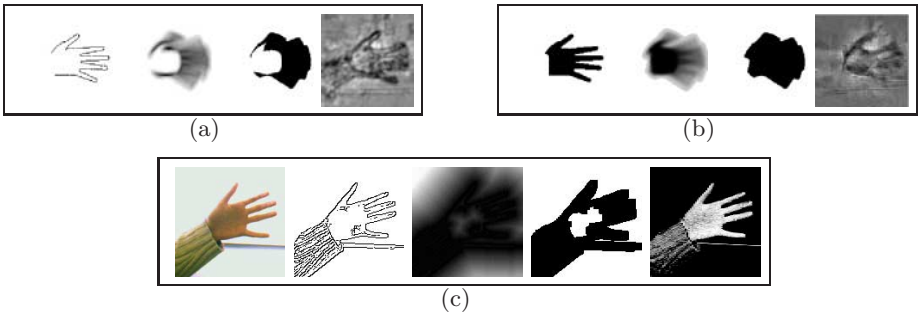


Fig. 3. Templates and Feature Maps used for Classification (a) Templates used for classifying edge features. From left to right: single template, marginalized template, binary marginalized template, template learnt from image data **(b)** Templates for classifying colour features, analogous to top row **(c)** Extracted features from an image with simple background. From left to right: input image, edges, distance transformed edge map, dilated edge map, colour likelihood image

constant. Note that when the distributions are fixed, the log-likelihood values for each colour vector can be pre-computed and stored in a look-up table beforehand. The corresponding templates **A** are shown in figure 3 (b).

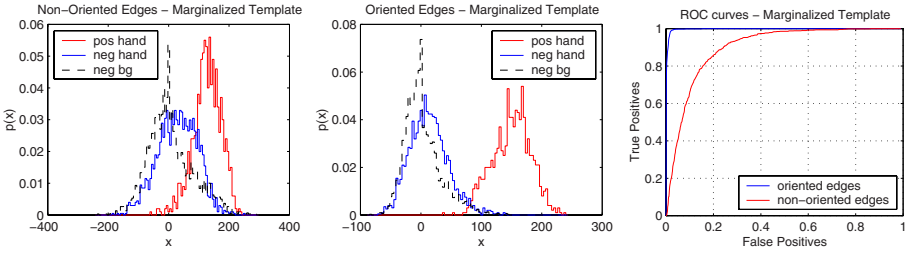
4 Results

In order to compare the performance of different classifiers, a labelled set of hand images was collected. Positive examples are defined as the hand being within a region in parameter space. For the following experiments this region is chosen to be a rotation of 30 degrees parallel to the image plane. Negative examples are background images as well as images of hands in configurations outside of this parameter region. The evaluation of classifiers is done in three independent experiments for different hand poses, an open hand, a pointing hand and a closed hand. The test data sets each contain 5000 images, of which 1000 are true positive examples. The classifiers are defined by the entries in the matrix **A**, described in the previous section, and illustrated in figure 3 (a) and (b).

4.1 Edge Templates

The following observations were made consistently in the experiments:

– In all cases the use of edge orientation resulted in better classification performance. Including the gradient direction is particularly useful when discriminating between positive examples and negative examples of hand images. This is illustrated in figure 4 (a) and (b), which show the class distributions for non-oriented edges and oriented edges in the case of marginalized templates with non-negative weights. The corresponding ROC curves are shown



(a) (b) (c)

Fig. 4. Including Edge Orientation Improves Classification Performance This example shows the classification results on a test set using a marginalized template **(a)** histogram of classifier output using edges without orientation information: hand in correct pose (red/light), hand in incorrect pose (blue) and background regions (black,dashed line) **(b)** histogram of classifier output using edges with orientation information. The classes are clearly better separated, **(c)** the corresponding ROC curve

in 4 (c), demonstrating the benefit of using oriented edges. This shift in the ROC curve is observed in different amounts for all classifiers and is shown in figure 5 (a) and (b).

- In all experiments the best classification results were obtained by the classifier trained on real data. The ROC curves for a particular hand pose (open hand parallel to image plane) are shown in figure 5. At a detection rate of 0.99 the false positive rate was below 0.05 in all experiments.
- Marginalized templates showed good results, also yielding low false positive rates at high detection rates. Templates using pixel-wise averaging and negative weights for background edges were found to perform best when comparing the three versions of marginalized templates. For this template the false positive rates were below 0.11 at a detection rate of 0.99.
- Using the centre template with chamfer or Hausdorff matching showed slightly lower classification performance than the other methods, but in all cases the false positive rate was still below 0.21 for detection rates of 0.99. Chamfer matching gave better results than Hausdorff matching, as can be seen in the ROC curve in figure 5 (b). It should be noted that this result is in contrast to the observations made by Huttenlocher in [8], for more details see [18].

The execution times for different choices of templates $\mathbf{A}$ were compared in order to assess the computational efficiency. Computing the scalar product of two vectors of size 128×128 is relatively expensive. However, the computational time can be reduced by avoiding the multiplication of zero valued entries in the matrix $\mathbf{A}$. For chamfer and Hausdorff matching, the template only contains the points of a single model projection. The number of points in the marginalized template depends on the size of the parameter space region it represents. In the experiments it contained approximately 14 times as many non-zero points as a single model template. When using a binary template the dot product

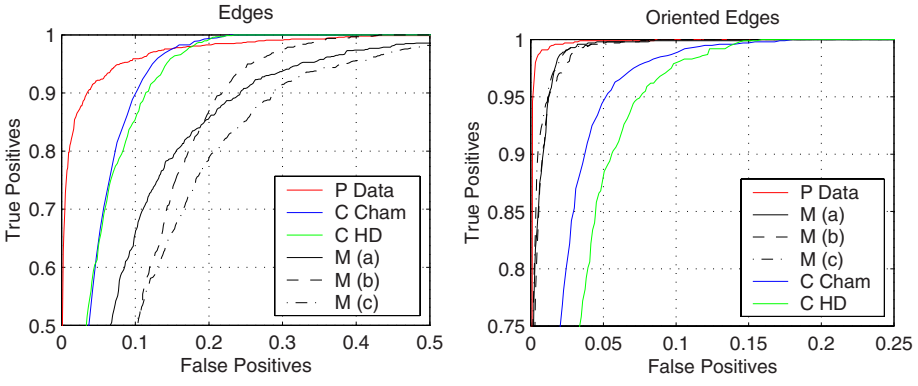


Fig. 5. ROC Curves For Classifiers This figure shows the ROC curve for each of the classifiers **(a)** edge features alone, and **(b)** oriented edges Note the difference in scale of the axes. The classifier trained on real image data performs best, the marginalized templates all show similar results, and chamfer matching is slightly better than Hausdorff matching in this experiment. When used within a cascade structure, the performance at high detection rates is important

computation simplifies to additions of coefficients. If both vectors are in binary form, a further speed-up can be achieved by using AND operations [18]. The execution times for correlating 10,000 templates are shown in table 1. The time for computing a distance transform or dilation, which needs to be only computed once for each frame when chamfer or Hausdorff matching is used, is less than 2 ms and is therefore negligible when matching a large number of templates. There clearly seems to be a trade-off between computation time and classification performance for the classifiers. When used in a cascaded structure, the detection rate of a classifier needs to be very high, so as not to miss any true positives. Chamfer and Hausdorff matching, while having a larger false positive rate, are about 10-14 times faster to evaluate than marginalized templates and about 40 times faster than the trained classifier.

4.2 Silhouette Templates

The same test data set as for edges was used, and the following observations were made:

– For the test set colour information helps to discriminate between positive examples of hands and background regions. However, there is significant overlap between the positive and negative class examples which contain a hand. Oriented edges are better features to discriminate between the hand in different poses, whereas colour features are slightly better at discriminating between the positive class and background regions.

Table 1. Computation times for Correlating Templates The execution times for computing the dot product of 10,000 image patches of size 128×128, where only the non-zero coefficients are correlated for efficiency, measured on a 2.4 GHz PC with Pentium IV. The last column shows the false positive rates for each classifier at a fixed detection rate of 0.99

Classification Method	Number of Points	Execution Time	fp at $tp = 0.99$
Chamfer	400	13 ms	0.10
Hausdorff	400	13 ms	0.12
Marginalized Template	5,800	186 ms	0.02
Binary Marginalized Template	5,800	136 ms	0.02
Trained Classifier Template	16,384	524 ms	0.01

– Both, centre template and marginalized template show better classification performance than the trained classifier, in particular in the high detection range. At detection rates of 0.99 the false positive rate for the centre template is 0.24, wheras it is 0.64 for the trained classifier. However, the trained classifier shows better performance at separating positive examples from negative example images containing hands. At a detection rate of 0.99, the false positive rate is 0.41 compared to 0.56 for the other two classifiers.

The evaluation can be performed efficiently by pre-computing a sum table, $\mathbf{B}^{sum}$, which contains the cumulative sums of costs along the x-direction:

$$\mathbf{B}^{sum}(x, y) = \sum_{i=1}^{x} \left(\log p^s(I(i, y)) - \log p^{bg}(I(i, y)) \right) \quad , \tag{2}$$

where in this equation the image I is indexed by its x and y-coordinates. p^s and p^{bg} are the skin colour and background colour distributions, respectively. This array only needs to be computed once, and is then used to compute sums over areas by adding and subtracting values at points on the silhouette contour. It is a convenient way to convert area integrals into contour integrals, and is related to the summed area tables of Crow [3], the integral image of Viola and Jones [22] or the integration method based on Green's theorem of Jermyn and Ishikawa [12]. Compared to integral images the sum over non-rectangular regions can be computed more efficiently, and in contrast to the technique of Jermyn and Ishikawa the contour normals are not needed. In contrast to these methods, however, the model points need to be connected pixels, e.g. obtained by line-scanning a silhouette image. The computation time for evaluating 10,000 templates is reduced from 524 ms to 13 ms for a silhouette template of 400 points. As the computation of the sum table $\mathbf{B}^{sum}$ is only computed once, this is negligible when matching a large number of templates.

Fig. 6. Hierarchical Detection of Pointing Hand Left: Input image, Next: Images with classification results super-imposed. Each square represents an image location which contains at least one positive classification result. Higher intensity indicates larger number of matches. Face and second hand introduce ambiguity. Regions are progressively eliminated, the best match is shown on the right

	Accepted				Rejected			
Level 1								
	9.84	11.08	12.21	12.29	22.90	22.90	22.90	22.90
Level 2								
	7.36	8.41	9.63	9.69	17.00	17.01	17.02	17.07
Level 3								
	6.68	6.69	6.85	6.90	12.21	12.24	12.25	13.21

Fig. 7. Search Results at Different Levels of the Tree This figure shows typical examples of accepted and rejected templates at levels 1 to 3 of the tree, ranked according to matching cost shown below. As the search is refined at each level, the difference between accepted and rejected templates decreases

4.3 Detection Examples

The tree-based detection method was tested on real image data. This corresponds to the initialization stage in the hierarchical filter [18, 19]. Figure 6 illustrates the operation of the classifiers at different levels of the tree. In this case the classifiers are based on oriented edges using the chamfer distance and skin colour silhouette. The classifiers at the upper levels correspond to larger regions of parameter space, and are thus less discriminative. As the search proceeds regions are progressively eliminated, resulting in only few final matches. The tree contains 8,748 different templates, corresponding to a pointing hand, restricted to rigid motion in a hemisphere. Figure 7 shows examples of accepted and rejected templates at different tree levels for a different input image.

5 Conclusion

In this paper the concept of a hierarchical cascade of classifiers for locating articulated objects was introduced. The classifiers can be obtained from a geometric

3D model or from training images. The motivation of this research has been ongoing work on hand tracking, in which we seek to combine the merits of efficient detection and tracking. Motivated by the success of tree-based detection, the parameter space is discretized to generate a tree of templates which can be used as classifiers. Even though their performance has been shown to be not as good as classifiers learnt from image data, they have the advantage of being easy to generate and being labelled with a known 3D pose, permitting their use in a model-based tracking framework.

Acknowledgements

The authors gratefully acknowledge the support of Toshiba Research. This work has also been supported in part by EPSRC, the Gottlieb Daimler–and Karl Benz–Foundation, the Gates Cambridge Trust, and the ORS Programme.

References

[1] V. Athitsos and S. Sclaroff. Estimating 3D hand pose from a cluttered image. In *Proc. Conf. Computer Vision and Pattern Recognition*, volume II, pages 432–439, Madison, USA, June 2003. 106, 107

[2] H. G. Barrow, J. M. Tenenbaum, R. C. Bolles, and H. C. Wolf. Parametric correspondence and chamfer matching: Two new techniques for image matching. In *Proc. 5th Int. Joint Conf. Artificial Intelligence*, pages 659–663, 1977. 106, 107

[3] F. C. Crow. Summed-area tables for texture mapping. In *siggraph*, volume 18, pages 207–212, July 1984. 113

[4] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley & Sons, New York, 2nd edition, 2001. 109

[5] P. F. Felzenszwalb. Learning models for object recognition. In *Proc. Conf. Computer Vision and Pattern Recognition*, volume I, pages 56–62, Kauai,USA, December 2001. 108, 109

[6] D. M. Gavrila. Pedestrian detection from a moving vehicle. In *Proc. 6th European Conf. on Computer Vision*, volume II, pages 37–49, Dublin, Ireland, June/July 2000. 106, 109

[7] N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to non-linear and non-gaussian bayesian state estimation. *IEE Proceedings-F*, 140:107–113, 1993. 105

[8] D. P. Huttenlocher. Monte carlo comparison of distance transform based matching measure. In *ARPA Image Understanding Workshop*, pages 1179–1183, New Orleans, USA, May 1997. 111

[9] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge. Comparing images using the hausdorff distance. *IEEE Trans. Pattern Analysis and Machine Intell.*, 15(9):850–863, 1993. 109

[10] D. P. Huttenlocher, J. J. Noh, and W. J. Rucklidge. Tracking non-rigid objects in complex scenes. In *Proc. 4th Int. Conf. on Computer Vision*, pages 93–101, Berlin, May 1993. 107

[11] M. Isard and A. Blake. Visual tracking by stochastic propagation of conditional density. In *Proc. 4th European Conf. on Computer Vision*, pages 343–356, Cambridge, UK, April 1996. 105

[12] I. H. Jermyn and H. Ishikawa. Globally optimal regions and boundaries. In *Proc. 7th Int. Conf. on Computer Vision*, volume I, pages 20–27, Corfu, Greece, September 1999. 113

[13] J. MacCormick and M. Isard. Partitioned sampling, articulated objects, and interface-quality hand tracking. In *Proc. 6th European Conf. on Computer Vision*, volume 2, pages 3–19, Dublin, Ireland, June 2000. 107

[14] C. F. Olson and D. P. Huttenlocher. Automatic target recognition by matching oriented edge pixels. *Transactions on Image Processing*, 6(1):103–113, January 1997. 106, 108

[15] S. Romdhani, P. H. S. Torr, B. Schölkopf, and A. Blake. Computationally efficient face detection. In *Proc. 8th Int. Conf. on Computer Vision*, volume II, pages 695–700, Vancouver, Canada, July 2001. 106

[16] H. Schneiderman and T. Kanade. A statistical method for 3D object detection applied to faces and cars. In *Proc. Conf. Computer Vision and Pattern Recognition*, volume I, pages 746–751, Hilton Head Island, USA, June 2000. 106

[17] G. Shakhnarovich, P. Viola, and T. Darrell. Fast pose estimation with parameter-sensitive hashing. In *Proc. 9th Int. Conf. on Computer Vision*, pages 750–757, Nice, France, October 2003. 106

[18] B. Stenger. *Model-Based Hand Tracking Using A Hierarchical Filter*. PhD thesis, University of Cambridge, Cambridge, U.K., 2004. 109, 111, 112, 114

[19] B. Stenger, A. Thayananthan, P. H. S. Torr, and R. Cipolla. Filtering using a tree-based estimator. In *Proc. 9th Int. Conf. on Computer Vision*, volume II, pages 1063–1070, Nice, France, October 2003. 105, 106, 107, 114

[20] A. Thayananthan, B. Stenger, P. H. S. Torr, and R. Cipolla. Learning a kinematic prior for tree-based filtering. In *Proc. British Machine Vision Conference*, volume 2, pages 589–598, Norwich, UK, September 2003. 105

[21] K. Toyama and A. Blake. Probabilistic tracking with exemplars in a metric space. *Int. Journal of Computer Vision*, pages 9–19, June 2002. 106

[22] P. Viola and M. J. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. Conf. Computer Vision and Pattern Recognition*, volume I, pages 511–518, Kauai, USA, December 2001. 106, 113

[23] O. Williams, A. Blake, and R. Cipolla. A sparse probabilistic learning algorithm for real-time tracking. In *Proc. 9th Int. Conf. on Computer Vision*, volume I, pages 353–360, Nice, France, October 2003. 105

Exploring Interactions Specific to Mixed Reality 3D Modeling Systems

Lucian Andrei Gheorghe[1], Yoshihiro Ban[2], and Kuniaki Uehara[1]

[1] Graduate School of Science and Technology, Kobe University
lucian@ai.cs.scitec.kobe-u.ac.jp
uehara@kobe-u.ac.jp
[2] Information Science and Technology Center, Kobe University
1-1 Rokko-dai, Nada, Kobe 657-8501, Japan
ban@kobe-u.ac.jp

Abstract. This paper proposes an interface especially tailored to create a design oriented realistic Mixed Reality (MR) workspace. An MR environment implies a natural mixture between virtual and real objects. We eplore the extension of real objects into the virtual world by having virtual objects attached to them. Also, in order to create a realistic environment, the problems of the occlusions between real and virtual objects are addresed. While providing more intuitive input methods than the conventional desktop modelers or Virtual Reality (VR) immersive systems, our prototype system enables the user to have an accurate perception of the shapes modeled. Thus, the user can recognize correctly the spacial location and real sizes of the models.

1 Introduction

Interaction techniques for Virtual Reality (VR) design systems have been widely explored and few Augmented Reality (AR) based systems are under development. Yet, a true Mixed Reality (MR) design environment has not emerged yet. This paper proposes an interface especially tailored to create a design oriented realistic MR workspace. In order to create such an interface several kinds of interactions should be treated. We categorize them as folows:

User - Virtual Objects Interactions further called UVO interactions, should enable the user to grab, translate, rotate virtual objects. Since the action will take place in a MR environment the virtual objects must be rendered to look as placed in the real world and the user should be able to use his hands in a very natural way, as if handling real objects. Furthermore, in order to provide means to choose between the functions provided by the system, a menu of some kind should be implemented.

CAD like Interactions further called CAD interactions, would allow the user to build virtual objects. Objects may be linked together, resized or reshaped. Here, we must stress our intention to keep the interaction activities as close to real ones as possible. So we experiment an implementation that does not

N. Sebe et al. (Eds.): HCI/ECCV 2004, LNCS 3058, pp. 117–128, 2004.

uses modeling tools. The user should be able to use his hands directly to grab and resize an object like in the case of an deformable real object.

User - MR Objects Interactions further called UMO interactions would allow interactions with MR Objects. The concept of Mixed Reality Object will be introduced as an real object with virtual extentions attached to it. Handling this kind of objects the user would experiment not only modeling VR objects but modeling real objects as well. This is one of the main features of our system and can only be realized in a robust mixed reality environment. To meet this requirement our system must handle two kinds of occlusion problems.

Hand - Virtual Objects Occlusions: The position of the user's hand must be determined. Then, the parts of the hand that are supposed to be occluded by virtual objects should be hidden and the other way around.

Real Objects - Virtual Objects Occlusion: Generating correct occlusions induced by all the real objects is a very difficult task, but the occlusions induced by any object used to build an MR Object must be adressed. In this case, at least a rough digital model of the object is needed. So, the real object must be introduced to the system by an easy to achieve, realtime and intuitive digitization process.

So as a main characteristic of our approach is the fact that would provide a "realistic workspace". That is, while providing basic CAD functions the differences between interaction with real and interactions with virtual objects are as small as possible. We focus on providing accurate input methods but also natural, intuitive ones.

A system that takes full advantage of these features can improve the trial-and-error process that takes place between modeling and the mass produc tion process in the industrial design field. That is, once new product being designed, for review needs, physical models (mock-ups) are build. When an change is needed, the basic CAD model is changed and a new model is build. Using our system, the user can compare newly developed models with existing products in a very natural and realistic way, by simply holding both of them. The sense of realness given by a MR system confirms the possibility of it replacing samples. Further, the modeling tools implemented enable changes of the model so any suggestions, ideas can be implemented "on the place" and verified in an efficient way, which is an great advantage over CAD and VR modeling systems.

2 Prototype System Configuration

Our system is built on the MR Platform (MRP) [9], which includes a parallax-less video see-through Head Mounted Display (HMD) and a software development kit. Using this Platform we were able to easily create the workspace presented in Figure 1 a). Modeling process will take place on a $120cm \times 100cm$ desk but it is not limited to this size. Several fiducials and one of the four receivers of a six Degrees-Of-Freedom (DOF) magnetic tracking sensors are used by the MRP to estimate the users head position and orientation. Like in any basic

a) workspace b) user view c) position sensors

Fig. 1. System outline

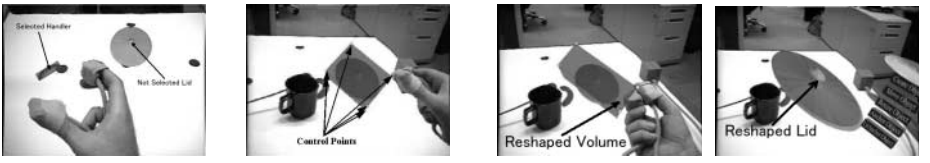
MR/AR system the images taken from the two video cameras placed in front of the users eyes are analyzed on a PC. The artificial objects are superimposed on these images. The result is displayed on the two screens in the HMD. MRP implements rendering functions too, so the superposition is done correctly and the user has the impression that virtual objects actually lie on the desk.
Figure 1 b) presents an example of an image projected on one of the two screens. The cups are real objects while the pot is a virtual object. Two tracking sensors are used for user hands movement recognition. The sensors are placed on the right hand's pointer and thumb. So these two fingers' positions can be tracked in real-time. Figure 1 c) presents the right hand with the two sensors installed. D is the distance between the thumb and the pointer. It's variations are used for gesture recognition.

3 UVO and CAD Interactions

At the moment the interaction with virtual objects includes **move**, **resize** and **menu selection**. These operations begin with an object selection step. For each object the gravitational center, and an enveloping volume are calculated. Figure 2 a) presents a selection case. The closest object to the hand, in this case the handler, is pointed by displaying its enveloping volume as a transparent parallelepiped. A grab action performed close enough to the object will trigger object selection.

Object movement means, naturally, that after a successful selection, the object will follow the users hand movements until it is released. The resemblance with the real world is maintained since the user needs to bring his right hand's fingers together as if picking something and spread them as if releasing some real object.

For **object resizing**, the eight vertexes of the enclosing volume of the selected object are used as control points. These points are displayed as small spheres like in Figure 2b). A grab gesture will be interpreted as a grab action on the closest point to the hand. The proximity condition is applied here too. Once grabbed, the control point can be moved and at the same time the enclosing volume will reshape as shown in Figure 2c).



a) object selection b) control point selection c) resizing an object d) object resized

Fig. 2. UVO and CAD Interactions

Each object has a coordinates system attached to it and all its vertexes are represented in this local coordinates system. The origin coincides with the enclosing volume's center and is defined by a transformation matrix, called here M_{object}. Now, the hand's absolute position at the release moment can be expressed like a vector as $P_{hand} = [p_x p_y p_z 1]$. The new position of the selected vertex coincides with the hands position, so it can be represented in the local coordinates system as in Equation 1.

$$V^{'} = P_{hand} \times M_{object}^{-1} \tag{1}$$

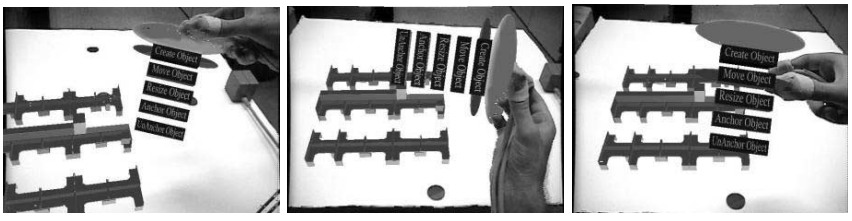
Considering the vertex initial position as $V = [v_x v_y v_z 1]$ a scale matrix that would move the vertex from V to $V^{'}$ can be calculated as in Equation 2.

$$M_{scale} = \begin{bmatrix} \frac{v_x^{'}}{v_x} & 0 & 0 & 0 \\ 0 & \frac{v_y^{'}}{v_y} & 0 & 0 \\ 0 & 0 & \frac{v_z^{'}}{v_z} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

Any zero values for v_x , v_y or v_z would mean a void object or, at most, a flat one so the scaling for that dimension is not performed. Using this matrix all the vertexes and dimensions of the object are recalculated in order to fit the new volume. Figure 2d) shows the result of a resize action. The lid takes the size of the new volume and reaches an unrealistic size.

Any modeling system must provide some kind of menu to enable functions selection. Given the fact that a well-designed MR environment should give the user good perception of positions and sizes we preferred a 3D widget type solution rather then VR "floating" menus [10]. Also passive-haptic feedback [11] is not a good solution since this would keep the user's hands occupied. Our menu is "built" out of right-angled parallelepipeds lined up vertically like in Figure 3. Initially the menu is floating above the desk. The user can move it around by grabbing it by the ellipsoidal part (Figure 3a)), so we named it **HandHeld-Menu**.

The menu will maintain its position and orientation relative to the user's hand so the user has the impression that it holds the menu (Figure 3b)). Releasing the menu makes it active, meaning that the user can select menu items. The menu remains in the same position while two halves of an ellipsoid will move up and down on the sides of these items pointing the element that is closest to the



a) menu in hand b) rotated menu c) menu selection

Fig. 3. HandHeld Menu

current hand position (Figure 3c)), as long as the hand is close enough to the menu. A grab event will be interpreted as a click on the currently selected item. In order to avoid wrong selections any click that is too far from the closest menu item is ignored.

4 UMO Interactions

The core of MR is mixing real objects with computer-generated objects. Our modeling process involves both types of objects so the user can compare sizes of artificial objects relative to real ones. For instance, in one test task, given a set of real cups, and a small virtual pot, the user is asked to resize the pot to a natural size, and set all the objects in a certain configuration. So in this case common knowledge regarding real objects can be used in handling artificial objects.

According to the lightning conditions of the room where the modeling process takes place, the layout of the virtual light sources is set. So, real objects and artificial objects will be illuminated in similar ways allowing better perception of positions.

We also implemented simple methods to resolve some of the occlusion problems. Performing video streaming and sensor input analysis, real time stereo 3D rendering, overall, the system should need a large amount of CPU time. So we tried to keep the occlusion analysis as simple as possible.

4.1 Hand-Virtual Object Occlusion

Figure 4 presents a few snapshots of different occlusion situations. The principle we used is basically finding the pixels that form the hand in the two images received from the video cameras and render them at the correct depth in the 3D scene.

The axis origin of the virtual world and the one of the sensors are the same. So a simple projection using the world matrix and the projection matrix used for the 3D rendering would be sufficient to determine the depth of the hand from the current point of view. The video input is two images of 640x480 pixels.

First the virtual objects are rendered on these images using MRP routines. Then, a copy of the original video input, half sized for speed considerations, is


a) hand occluding virtual objects


b) partial occlusion


c) virtual object occluding hand

Fig. 4. Occlusion situations

processed as follows.All the pixels that are not part of the hand, or the ones that are but the value of the depth buffer of the already rendered scene is smaller then the hand's, are erased. The remaining image is double size rendered on top of the scene (without depth check).

Notice here, that all the pixels are rendered at the same depth. Since the thumb is closer to the palm in most of the gestures during the modeling process, it's depth is used. The deployment of a physical model of the hand will enable a more detailed rendering method and is currently under development.

4.2 Real Object - Virtual Object Occlusion

At the moment, only occlusions produced by one real objects on virtual objects are treated. We implement a classic solution where a virtual model of the real object is used to occlude virtual objects. In many cases, for tracking means, CAD models resulted from a digitization process, or predefined ones are used. The incompatibility of these methods with our system is that in any case the model needs to be imported at the initialization stage of the system so only previously known objects can be used.

One of the main features of our interface is that any real object can be mixed with virtual ones. An easy to be performed digitization process is deployed so models for any object can be built in no-time and the object is so introduced to our system. In order to stay close to our commitment to a natural interface we inspired from the human behavior. When one can not see an object (a blind person, one in the dark etc), one tries to understand the shape of the object by simply touching it repeatedly (palpation).

a) palpation process b) occlusion effect

Fig. 5. Digitization Process

Figure 5a) presents an user digitizing a cup. One magnetic sensor being placed on the real object, the user will slide his right hand's fingers on the surface of the object just like a blind person would do in order to memorize a new object.

The position of the sensors on the right hand relative the one on the object are recorded. Notice here, that as long as the user keeps his right hand fingers on the object, he can manipulate it freely with the left one which gives this method a certain amount of naturalness. The small spheres in Figure 5a) represent the measured vertexes. The small error margin provided by the sensors guarantees a good accuracy level for these measurements.

As a result of this measurement process a set of vertexes placed mostly on the surface of the object is obtained. This points can be used to rebuild the surface. Many algorithms for this purpose have already been developed. We use the implementation provided by the VisualisationToolkit [19] library for the Ph D work of Hugues Hopp [20]. A signed measure of the distance to the surface is computed and sampled on a regular grid. The grid can then be contoured at zero to extract the surface. The default values for neighborhood size and sample spacing give reasonable results for most uses so adjustments are not needed. The reconstructed surface will be transparent, but, when rendered in the same place with the real object, it will correctly occlude parts of the virtual object, working like a complex stencil buffer.

Now, it must be said that even though the surface reconstruction algorithm is a robust one, there is a limitation induced by the fact that it gives good results only when enough sample points are provided. So when in some area insufficient information is given faulty surfaces are produced. Figure 6a) presents a faulty surface induced by the lack of density in vertexes. Figures 6b) and 6c) and present correctly digitized objects.

To deal with this inherent problem two methods are currently under development. One way is to let know the user that more samples are needed in certain area. But since the shape of the object is not known it is difficult to estimate this kind of areas. So we are developing a procedure that will change the color of the sampled point that have too few neighbors. The points on the edges or corners may be faultily chosen but the user can simply ignore them and take

a) faulty surface b) cup body c) pot body

Fig. 6. Digitization Surfaces

this color change only as an advisory indication. Another method that will be experimented is deleting polygons that are faultily produced.

The object obtained will be automatically anchored to the sensor on the real object. Due to the nature of the digitization process alignment of the virtual model and the real object are correct and only small errors are induced by the magnetic sensor. Now, since this virtual object moves along with the position sensor placed on the real object, and the 2 worlds are mapped 1 to 1, the real object will seem to occlude virtual objects as if being in the same space. In Figure 5b) the hand and the cup are closer to the user than the virtual pot so both hide it correctly.

These features give the user a good position and size perception. So even though, in objects or control point selection the closest to hand rule is applied, a strict distance condition is also in place. This rule can be applied only when the user can have a good enough position perception. Otherwise most of the selection actions will fail. The reinforcement of this rule also contributes to the realism of the scene. If any grip event would mean that the closest object will start floating at a constant distance from the hand, or that an object will jump into the user's hand, then the realism of the environment will suffer.

4.3 MR Object

We imagined the user as being a bridge between the real and virtual world. The right hand can enter the virtual one, while the left one stays in the real world. One modeling process will be a combination of actions of the right hand in the virtual world and the left one in the real one. A real object can be augmented with virtual objects, we call this a "mixed reality object". A 4th position sensor is placed on this object. Then, an anchor action would mean that a virtual object would be linked to the real object. Figure 7 shows a a real cup which has a real handler, a virtual one and a virtual lid.

In order to keep the virtual object in the same position relative to the real object to which it is connected, the position of the right hand relative to the sensor on the real object is used.

Considering M_{hand} as the transformation matrix defining the right hand position, and M_{object} the matrix corresponding to the position of the sensor

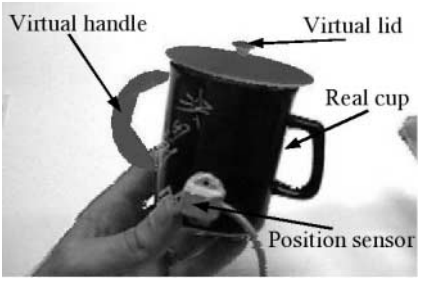


Fig. 7. Mixed reality cup elements

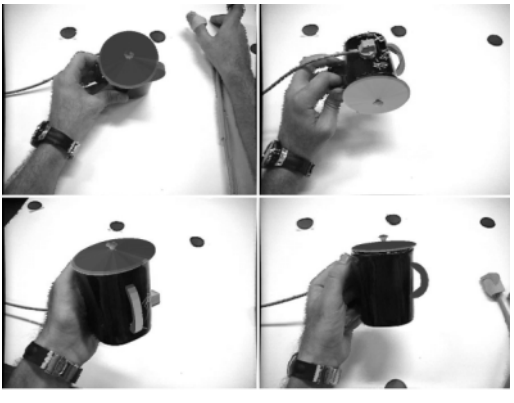


Fig. 8. Movement snapshots

on the real object, Equation 3 defines a relative transformation matrix, called here M_{anchor}.

$$M_{anchor} = M_{object}^{-1} \cdot M_{hand} \tag{3}$$

During an "Anchor" operation the value of M_{anchor} is set in the moment the virtual object is released, so the virtual object position is $M_{virtualObject} = M_{hand}$ at that moment. Now, since the position sensor maintains its position on the real object, then if the position of the virtual object relative to the sensor is constant, also its position relative to the real object is constant. So M_{anchor} can be used to recalculate the position of the virtual object when the real one has been moved to a new position, M'_{object}. The new position, $M'_{virtualObject}$ is calculated as in Equation 4.

$$M'_{virtualObject} = M'_{object} \cdot M_{anchor} \tag{4}$$

This way, whenever the real object will be moved the virtual one will move along maintaining it's relative position. Figure 8 presents a few snapshots of the user's view while handling the mixed reality cup.

Observe how the virtual lid and handler follow the real cup. The user gets the impression that the virtual objects (lid and handler) and the real object (the cup) are connected, forming a "mixed reality object".

5 Conclusions and Future Works

We presented the preliminary results of a project developing a Mixed Reality 3D modeling system which makes possible the creation of 3D models within an MR system. While providing more intuitive input methods than the conventional desktop modelers or VR immersive systems, our system enables the user to have an accurate perception of the shapes modeled, appreciate correctly the spatial location and real size. Occlusion problems are partially addressed with a lightweight solution. We introduced the concept of mixed reality object that has a real part (a real object) and a virtual part (virtual objects).

While not being throughly tested yet, a simple modeling task test revealed encouraging feedback. MR experienced and unexperienced users have been asked to put together the parts (lid, body, neck and handle) of a pot and resize it to a natural size compared to a set of real cups. Handling of objects and linkage received high reviews. The resizing operation was characterized in most cases as "fairly easy to perform". This is caused mainly by the size of the sensors . This size also generates some limitations for the digitization process, but at the moment only a rough model is sufficient. Both issues are and currently addressed Position perception received good reviews in the case of mixed reality objects modeling. Users have also been asked to add a lid and a handler to a cup like in Figure 7. Since the real object and the virtual object are hold in the left hand and right hand respectively, the anchoring operation could be easily performed.

Comparison with the interfaces implemented in modeling systems that received high reviews reveal the main improvements generated by our implementation. [13] provides a multitude of functions. The user can build and edit models with a high accuracy level due to the implementation of a grid and several input constraints. But in order to achieve some simple task the user has to make use of specialized tools. For example for rotating an object the user has to pick up the "Rotator" tool from the tools shell and than select the object to be rotated. In our implementation we eliminate the need for such tools and enable the user to make these simple operations in a more natural way.

[12] also provides an interface that allows high precision modeling and a validation method in an AR environment. But since for the modeling interface the stylus/tablet paradigm is implemented, physical tools need to be changed in order to make full use of the system. Even more, the visualization hardware needs to be changed between the modeling phase and the testing phase. The use of our interface enables modeling in a MR environment directly so the modeling/testing/unification of real objects and artificially ones can take place more smoothly and the need to change physical tools disappears.

Finally, it must be said that hands movement has been treated widely by several researchers ([6]. [7], [8] etc). There are some similarities between the

grab action we implemented and the pinch gloves paradigm [7]. While providing a much wider set of operations, a data glove still needs at least one magnetic sensor in order to become a 3D input device. Considering this, the fact that for our system only one type of pinching action is sufficient, and also the fact that connecting two magnetic sensors straight on the fingers will provide more accurate measurements we chose placing two magnetic sensors on the dominant hand's thumb and pointer. From this point of view, this paper does not present a completely new recognition algorithm, rather it presents the results of a simple and pragmatic implementation.

In terms of future works, one line under development is the modeling functions available for virtual objects. Creation of new objects using rectangles, spheres, 16 control points curves, polygonal shapes. rotation objects will be available. Using the control points technique for resizing, reshaping can be implemented. Grabbing vertexes and changing their position can be a realistic option.

But the area which seams to be most attracting for explorations is the mixed reality objects one. This experiment will be continued and at least one improvement is under development at the moment. First a method to make distinction between the right hand and left hand must be developed. Using the fact that the right hand position is known, a labeling step can be a solution. Then, an image processing method is bound to replace the two sensors on the right hand. At the same time this will enable a more precise implementation for hand induced occlusion problems.

References

[1] F. Brooks. What's real about virtual reality. In: IEEE Computer Graphics and App., vol. 19, no. 6, November 1999.

[2] H. Iwata. Feel-through: Augmented Reality with Force Feedback. In: Mixed Reality, pp. 215-227 Omsha, Japan, 1999.

[3] M. Aizawa and K. Hayashibe. The system for designing layout of indoor space using virtual reality. In: Proceedings of the VRSJ Fifth Annual Conference 2002, pp. 25-28, 2002.

[4] M. Foskey, M. A. Otaduy and M. C. Lin. ArtNova: Touch-Enabled 3D Model Design. In: Proceedingss of IEEE Virtual Reality 2002, pp. 119-126, 2002.

[5] J. Lee, G. Hirota and A. State. Modeling Real Objects Using Video See-through Augmented Reality. In: Proceedingss of ISMR 2001, pp. 19-26, 2001.

[6] D. A. Bowman and C. A. Wingrave. Design and Evaluation of Menu Systems for Immersive Virtual Environments. In: Proceedings of IEEE IVR 2001, pp. 149-156, 2001.

[7] J. S. Pierce, B. C. Stearns and R. Pausch. Voodoo Dolls: Seamless Interaction at Multiple Scales in Virtual Environments. In: Proceedings of the 1999 Symposium on Interactive 3D Graphics, pp. 141-145, 1999.

[8] L. Joseph and R. Zeleznik Flex and Pinch: A Case Study of Whole Hand Input Design for Virtual EnvironmentInteraction. In: Proceedings of the Second IASTED International Conference on Computer Graphics and Imaging,, pp.221-225, 1999.

[9] S. Uchiyama, K. Takemoto, K. Satoh, H. Yamamoto and H. Tamura. MR Platform: A basic Body on Which Mixed Reality Applications Are Built. In: Proceedings of IEEE ISMAR 2002, pp.246-256,2002. 118
[10] L. Cutler, B. Frohlich and P. Hanrahan. Two-Handed Direct Manipulation on the Responsive Workbench. In: Symposium on Interactive 3D Graphics, pp. 107-114, Providence, 1997 . 120
[11] R. W. Lindeman, J. L. Sibert and J. N. Templeman. The Effect of 3D Widget Representation and Simulated surface Constraints on Interaction in Virtual Environments. In: Proceedings of IEEE VR 2001,pp.141-148, 2001. 120
[12] M. Fiorentino, R. de Amicis, G. Monno and A. Stork,. Spacedesign: A Mixed Reality WorkSpace for Aesthetic Industrial Design. In: Proceedings of ISMAR 2002,pp.86-97, 2002.
[13] K. Kiyokawa, H. Takemura and N. Yokoya. SeamlessDesign for 3D Object Creation. In: Mutimedia Computing and Systems January-March 2000,pp.22-33, 2000.
[14] SensAble Technologies Inc. FreeFormTM modeling system. Available at: http://www.sensable.com/products/3ddesign/freeform/index.asp, 1999.
[15] G. Roth and E. Wibowo. An Efficient volumetric method for building closed triangular meshes from 3-D image and point data. In: Graphics Interface 97, pp.173-180, 1997.
[16] H. Tamura et al. Mixed reality: Future dreams seen at the border between real and virtual worlds. In: IEEE Computer Graphics and Applications, vol.21, no.6, pp.64-70, 2001.
[17] M. Fjeld, K. Lauche, M. Bichsel, F. Voorhorst, H. Krueger and M. Rauterberg. Physical and Virtual Tools: Activity Theory Applied to the Design of Groupware In: Computer Supported Cooperative Work, no.11, Kluwer Academic, Netherlands, 2002.
[18] D. A. Bowman, D. B. Johnson, L. F. Hodges. Testbed Evaluation of Virtual Environment Interaction Techniques. In: Presence, vol.10, no.1, pp.75-95, Massachusetts Institute of Technilogy, 2001. Visualization ToolKit.
[19] Kitware Inc. Visualization ToolKit. Available at: http://www.vtk.org/, 2004.
[20] H. Hoppe. Surface reconstruction from unorganized points. Available at: http://www.research.microsoft.com/ hoppe, 2004.

3D Digitization of a Hand-Held Object with a Wearable Vision Sensor

Sotaro Tsukizawa, Kazuhiko Sumi, and Takashi Matsuyama

Graduate School of Informatics, Kyoto University
Sakyo Kyoto 606-8501, Japan
{tsucky,sumi}@vision.kuee.kyoto-u.ac.jp
tm@i.kyoto-u.ac.jp

Abstract. It is a common human behavior to hold a small object of interest and to manipulate it for observation. A computer system, symbiotic with a human, should recognize the object and the intention of the human while the object is manipulated. To realize autonomous recognition of a hand-held object as well as to study human behavior of manipulation and observation, we have developed a wearable vision sensor, which has similar sight to a human who wears the sensor. In this paper, we show the following results obtained with the wearable vision sensor. First, we analyzed human manipulation for observation and related it with acquirable visual information. Second, we proposed *dynamic space carving* for 3D shape extraction of a static object occluded by a dynamic object moving around the static object. Finally, we showed that texture and silhouette information can be integrated in vacant space and the integration improves the efficiency of *dynamic space carving*.

1 Introduction

Human-computer symbiotic systems have been a popular subject of study in resent research. Symbiotic relationships between a person and a computer will require recognition both of the target object and the person. Ideally, the object should be recognized while it is in the person's hand. The feeling of the held object should be recognized too. Although, the person and the object can be observed by environmental sensors, we consider a wearable system because the computer system can share similar sight with the person. In other words, such a computer system can share the same experience with a human. Using a wearable vision system, it is easy to obtain similar visual information with the person including the object and the person's body. Fig.1 shows the wearable vision sensor we have developed. It is equipped with a gazing direction detector, which we refer to as the eye-mark recorder, and a stereo pair of cameras, each of which can control pan, tilt, and zoom. Manipulating an object in hand for observation is a kind of hand-held action. So far, various research has been conducted on holding [1][2]. Most of the research is concerned with hand-object relationships. Hand-object-view relationships are not studied yet. For recognizing the object and understanding the person's will, what kind of view can be acquired is the

N. Sebe et al. (Eds.): HCI/ECCV 2004, LNCS 3058, pp. 129–141, 2004.

Fig. 1. Wearable Vision Sensor

biggest concern. Introducing the idea of "view" into hand-object relationships will open a new possibility to estimate the person's feelings, such as interest and intent concerning the object in hand. However, the main research target will be the 3D shape extraction both of the object and of the hand. In this paper, we analyze and classified the hand-object-view relationships into four typical cases and show what kind of visual information can be acquired from these cases.

For 3D shape reconstruction, there have been two major approaches. One is to use multiple synchronous images taken by multiple cameras [3]. The other is to use multiple asynchronous images taken along a time sequence. The synchronous approach can deal with a dynamic scene, however, the asynchronous approach assumes a static scene in which nothing changes along the time sequence and the images are equivalent to the synchronous approach. Because of this similarity, we can apply well studied 3D reconstruction methods like the factorization methods [4], volume intersection [5], and space carving [6]. However, a hand-manipulated object can also be obscured by the person's hands. Since each hand during manipulation changes its shape and location relative to the object dynamically, it is not treated by the asynchronous single camera approach. Although, the synchronous approach can manage this problem, it is not suitable for wearable vision. The 3D shape extraction of a hand-held object by wearable vision is a new computer vision problem.

In this paper, 3D shape extraction of a hand-held object is regarded as a new class of shape extraction problems from asynchronous images which are captured in the situation where a dynamic occluding object exists.

The approach we propose is based on *vacant space*. *Vacant space* is defined as the space that is certain not to be occupied by any object. It can be derived both from silhouettes and from texture. Since the hand is a dynamic object occluding the static object inside, the *vacant space* will change from image to image, and extend its space until it reaches the boundary of the static object. Finally, we can get the 3D shape of the static object without the dynamic occluding object. The contribution of this paper is as follows:

1. From the observation viewpoint, we analyze human manipulation for observation, and we show that a silhouette and a texture have a complementary relation.
2. We propose *dynamic space carving* for 3D shape extraction of a static object occluded by a dynamic object moving around the static object. We show that by using *vacant space*, the dynamic object will be eliminated along the carving.
3. We show that texture information and silhouette information can be integrated in *vacant space* and the integration improves the efficiency of *dynamic space carving*.

The composition of this paper is as follows. Section 2 describes our approach and the classification of relationships between the manipulation type and the visual information. Section 3 describes the technique of 3D shape extraction of a hand-held object by using silhouette information and texture information from asynchronous multiple viewpoint images. In Section 4, we evaluate our approach with real images. Finally, in Section 5, we summarize our approach.

2 Our Approach

2.1 Vacant Space

In this research, time series images captured with the wearable vision sensor are asynchronous multiple viewpoint images. In an object-centered coordinate system, this problem can be defined as a 3D shape extraction problem from asynchronous images in the situation where a dynamic occluding object exists.

It is not always easy to segment an object and a hand when the object is in the hand. Rusinkiewicz reconstructed a hand-held object in real time [11]. He made detection of a hand impossible by gloving a holder's hand. Since to glove and manipulate an object is unnatural for a human, the approach is effective in a laboratory, but is not suitable to use in everyday life. Therefore, we don't use a glove, and the object and the hand are observed as a combined object. The shape of the combined object is changing along time. This makes it difficult to apply conventional techniques such as shape from silhouettes [7] and space carving [6], because these techniques depend on correspondence on the stable object texture. Instead, we propose to detect *vacant space*. Since the hand moves but the object doesn't move in the object-centered coordinate system, if *vacant space* is carved, the space which a hand occupies will intersect that of the moving hand. The intersection will become zero if the hand moves in a sufficiently large area. On the other hand, the object space will never become *vacant space*. So, if *vacant space* is carved for a long time, only the object remains in space.

There is silhouette information and texture information available as information acquired from an image captured with the wearable vision sensor. In each viewpoint, because of the silhouette constraint [8], a hand and an object surely exist in a visual cone obtained from a silhouette. Therefore, the outside of the visual cone is *vacant space* obtained by silhouette. In addition, if we can

(a) Shape Acquisition (b) Visual Texture Acquisition (c) Haptic Texture Acquisition (d) Total Appearance Acquisition

Fig. 2. The Classification of the Relationships

obtain a depth map by template matching and stereo analysis, the space from the viewpoint to the foreground of the depth map is *vacant space* obtained by texture, too.

2.2 The Classification of Hand-Object-View Relationships

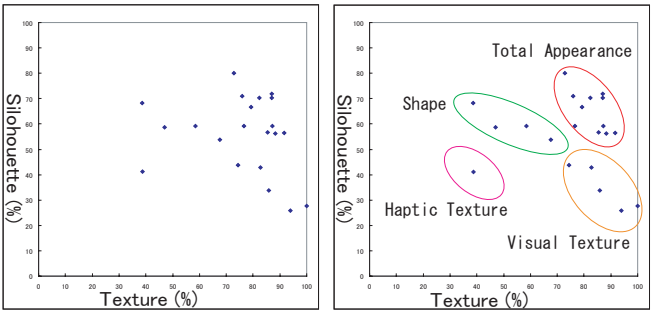
In this subsection, we discuss the silhouette information and texture information in the image of a hand-held object. When a person takes an object in their hands and observes it, the person cannot acquire all of the information of the object simultaneously. Thus, the person manipulates the object to acquire necessary information. When the person manipulates it, the object's visible part changes depending on a type of holding, or the physical relationship of the object and the hand seen from the person's viewpoint. We classify the hand-object-view relationships into four classes according to the information which the holder of the object can acquire.

Shape Acquisition :
When a person observes an object's shape, the person's viewpoint, the object, and their hand are located in a line. In the captured image, because of occlusion, a small part of the object's texture is obtained, but mostly the object silhouette is obtained as shown in Fig.2 (a).

Visual Texture Acquisition :
When a person observes an object's visual texture, seen from the person's viewpoint, their hand is on the object's backside. In the captured image, mostly the object's texture is obtained, but small part of the object's silhouette is obtained as shown in Fig.2 (b).

Haptic Texture Acquisition :
When a person observes an object's haptic texture, their hand occludes most of the object. In the captured image, a small part of the object's texture and the silhouette is obtained as shown in Fig.2 (c).

Total Appearance Acquisition :
When a person pinches an object to observe total balance of the object, seen from their viewpoint, their hand barely touches the object covering little



(a) Correlation Diagram (b) Classification

Fig. 3. Silhouette-Texture Correlation

important information. In the captured image, mostly the texture and the silhouette is obtained as shown in Fig.2 (d).

In order to confirm the classification, we developed the following experiment. We captured a scene that a person manipulated a cup to observe it. We captured 10 second and obtained 20 images. Fig.3 (a) shows that correlation of ratio of the object's acquirable silhouette and ratio of the object's acquirable texture in one of the images. The ratio of its silhouette is the ratio of part which appeared as a contour of a silhouette to the object's contour in an image. And the ratio of its texture is the ratio of non-covered part of the object's texture to the object's part in one of the images. The correlation coefficient is -0.31 and there is a negative correlation between them. It means that a possibility that a silhouette can be acquired is high if it is hard to acquire texture, and a possibility that texture can be acquired is high if it is hard to acquire a silhouette. And this experimental result implied that hand-held action can be classified into the four classes as shown in Fig.3 (b). Since information obtained from a captured image changes with the class, ideally a computer system should change information used for reconstruction. But it is difficult for a computer to identify the class of the image. Therefore, it is appropriate to use both silhouette information and texture information in hand-held action.

On the other hand, although it is impossible to obtain a silhouette of an object's concave part, the texture of the part can be obtained. Therefore, the part may be reconstructed by using texture information. And, although it is impossible to obtain an appropriate texture from a periodic texture or a monotonous texture, the silhouette of the part can be obtained. Therefore, the part may be reconstructed by using silhouette information.

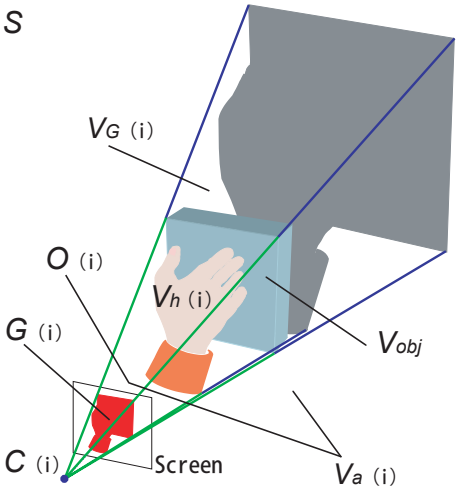


Fig. 4. The Parameters in Reconstruction Space

Thus, silhouette and texture have a complementary relation. Therefore, using both is appropriate, and we use both to detect *vacant space*.

3 Dynamic Space Carving

3.1 Vacant Space Detection

Conventional 3D shape reconstruction approaches try to calculate the space of the object or the surface of the object. However in this problem under occlusion by dynamic object, these approaches produce multiple inconsistent results caused by the dynamic object. Instead, we focus on the space, which is occupied neither by the static object nor by the dynamic occluding object. We refer to this space as *vacant space*. It can be calculated both from silhouette information and from texture information. *Vacant space* grows as we add new viewpoints and as the dynamic object changes its shape, but does not extend beyond the surface of the static object. Finally, we get the shape of the static object.

3.2 Definition of Parameters

We assume that images are captured from time t_1 to time t_n. Before applying our approach, the camera motion is recovered from feature tracking of the object. See 4.1 for camera motion recovery. In this paper, we use the following notations as shown in Fig.4.

S : Space of an object-centered coordinate system
V_{obj} : Space where the object occupies
$V_h(i)$: Space where a hand occupies at time t_i
$C(i)$: A viewpoint at time t_i
$G(i)$: A silhouette of V_{obj} and $V_h(i)$ obtained in viewpoint C_i
$V_G(i)$: Space that is back-projected $G(i)$ from viewpoint $C(i)$ (visual cone obtained by $G(i)$)
$O(i)$: Space from viewpoint $C(i)$ to the foreground of a depth map obtained in $C(i)$
$V_a(i)$: *vacant space* at time t_i

At time t_i, *vacant space* $V_a(i)$ is defined in Eq.(1)

$$V_a(i) = O(i) \cup \overline{V_G(i)} \tag{1}$$

And we define $P(p, V_x)$ as the probability that a voxel p is included in space V_x. In this research, the following conditions shall be fulfilled for S, V_{obj}, and $G(i)$.

- S and V_{obj} are contained in the sight of the camera in all the viewpoints $C(1), C(2), \ldots, C(n)$.
- Silhouette $G(i)$ is obtained exactly.

3.3 Reconstruction by Dynamic Space Carving

We extract the hand-held object space by carving *vacant space*. We call this technique *dynamic space carving*.

The space where the object does not exist certainly is described as a set of voxels p ($p \in S$) where the proposition Eq.(2) is true.

$$\bigcup_{i=1}^{n} \{p \in V_a(i)\} \tag{2}$$

Therefore, to prove that a hand is not included in an extracted shape, we should just show that the probability that p ($p \in \overline{V_{obj}}$) is not included in $V_a(i)(i = 1, 2, \ldots, n)$ in all viewpoints becomes zero.

We suppose that a person manipulates an object for a long time ($n = \infty$), and images are captured in a sufficient number of viewpoints. If it is supposed that a hand moves randomly, the space excluding the object in a viewpoint C_i, can be described in probability theory by whether a voxel p ($p \notin V_{obj}$) fulfills Eq.(3) or not.

$$p \in \overline{V_a(i)} \tag{3}$$

Because Eq.(4) can be derived from our definitions, Eq.(5) can be concluded.

$$\overline{V_a(i)} \subset S \tag{4}$$

$$P(p, \overline{V_a(i)}) \leq P(p, S) \tag{5}$$

Therefore, since Eq.(6) and Eq.(7) are similarly derived from our definitions, Eq.(8) can be concluded.

$$P(p, S) = 1 \tag{6}$$

$$S \cap \overline{V_a(i)} \neq \phi \tag{7}$$

$$P(p, \overline{V_a(i)}) < 1 \tag{8}$$

The probability E that p ($p \in \overline{V_{obj}}$) is not included in $V_a(i)(i = 1, 2, \ldots, n)$ in all viewpoints is defined in Eq.(9).

$$E = P(p, \overline{V_a(1)})\ P(p, \overline{V_a(2)}) \cdots P(p, \overline{V_a(n)}) \tag{9}$$

Eq.(10) is derived from Eq.(8) and Eq.(9).

$$\lim_{n \to \infty} E = 0 \tag{10}$$

Therefore, if a hand moves randomly and is captured for a long time, the probability becomes zero that the voxel p which is not included in the object $\left(p \in \left(S \cap \overline{V_{obj}}\right)\right)$ is not included in $V_a(i)$. Thus, the object shape which does not include the hand can be extracted by *dynamic space carving*.

4 Experimental Results

In order to evaluate the proposed technique, we developed the following experiments. First, to check the fundamental effectiveness of the proposed technique and to evaluate errors, we reconstructed a simple shape object. Second, to check that the technique can apply to a common object, we reconstructed a toy figure of a monster.

4.1 The Flow of Processing

We extracted the 3D shape of a hand-held object by the following procedures.

1. **Capture**
   We captured images in the situation of manipulating an object with a wearable vision sensor.
2. **Camera Motion Recovery**
   To recover the camera motion, we estimated the position of distance data. First, we extracted feature points with the Harris Corner Detector [9], and we obtained 3D feature points by stereo analysis in each image. Next, we estimated the camera position by using the advanced Iterative Closest Point algorithm [10], and we recovered the camera motion.

3. **Depth Map Acquisition**
   In each viewpoint, we match points between stereo pair images by coarse-to-fine template matching [12], and we obtained a depth map and detected vacant space by stereo analysis.
4. **Silhouette Acquisition**
   By background subtraction, we obtained a silhouette containing the object and the hand in each viewpoint. And we obtained vacant space from the silhouette.
5. **Dynamic Space Carving**
   We carved the vacant space by the technique shown in the previous section.

4.2 Evaluation with a Simple Shape Object

In the situation that a person manipulated a random patterned photo frame, we captured it with the wearable vision sensor at 1 frame per second as shown in Figs.5 (a), (b), (c), (d). We obtained asynchronous images with the wearable vision sensor in 22 viewpoints, and we extracted the 3D object shape from these asynchronous images. The result extracted only by using the silhouette information is Fig.6 (a), and the result extracted by using the silhouette and texture information is Fig.6 (b). We compared those result models with the ground truth as shown in Fig.6 (c). The results are plotted in Fig.7 with graphs (a) and (b) showing respectively, the number of extra voxels, and missing voxels compared with the ground truth which is calculated from the geometrical facts. Extra voxels means voxels that the extracted model includes but ground truth does not, and missing voxels means voxels that the extracted model doesn't include but that the ground truth includes. The error ratios of final results to number of ground truth voxels is as shown in Table 2. The time which this processing required is shown in Table 1.

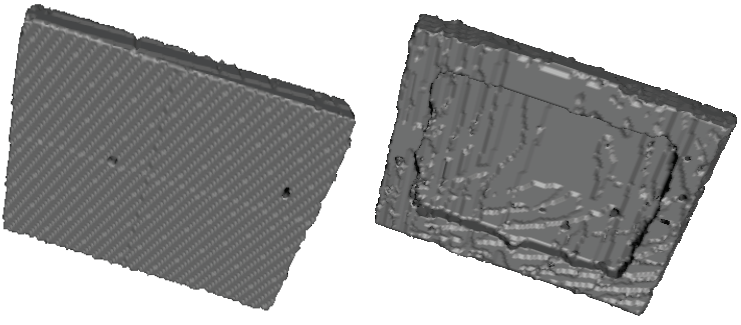
In the experimental results, when we extract the object shape by using silhouette information with many viewpoint images, we can reconstruct the 3D object shape which is not included the hand. And, by using silhouette and texture information, we can reconstruct it with almost the same accuracy with fewer viewpoints, namely, by capturing for a short time. Moreover, we can reconstruct the concave part which is impossible to reconstruct only by using silhouette information. In the result by using silhouette and texture information, there are more missing voxels than the result by using only silhouette information. But compared with extra voxels, missing voxels are few. Therefore, it can be said that to use silhouette and texture information is effective for the 3D reconstruction problem of a hand-held object.

4.3 Evaluation with a Complex Real Object

In the situation that a person manipulated a toy figure of a monster as shown in Fig.8 (a), we captured it with the wearable vision sensor. First, we obtained 13 asynchronous multiple viewpoint images. Next, we obtained vacant space, and we extracted the figure shape by carving the vacant space. The extracted figure

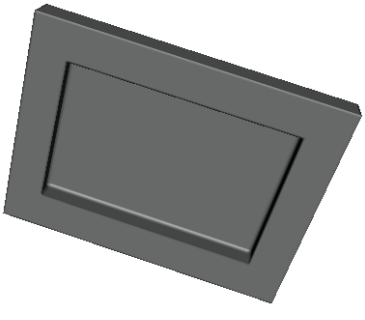
(a) 0 sec (b) 8 sec (c) 10 sec (d) 18 sec

Fig. 5. Captured Images of Photo Frame



(a) Extracted Shape by Silhouette Information

(b) Extracted Shape by Silhouette & Texture Information



(c) Ground Truth

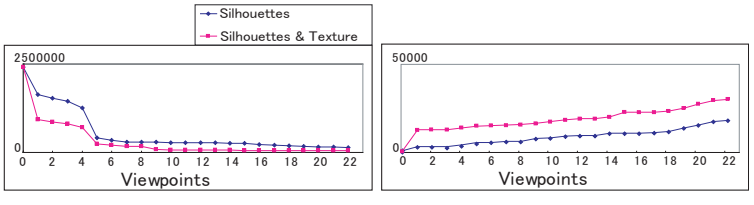
Fig. 6. Photo Frame Data Set

Table 1. Processing Time

	Time (sec)	
	Silhouettes	Silhouettes & Texture
Capture	22	22
Camera Motion Recovery	43	43
Depth Map Acquisition	none	248
Silhouette Acquisition	4	4
Dynamic Space Carving	48	42
Total	117	359

Table 2. Error Ratios to Number of Ground Truth Voxels

	Silhouettes	Silhouettes & Texture
Extra Voxels	86.8 %	31.0 %
Missing Voxels	11.0 %	18.6 %



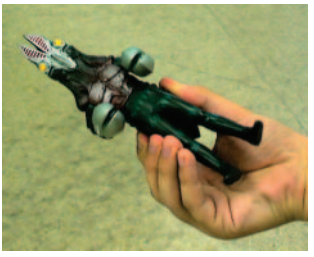
(a) Extra Voxels (b) Missing Voxels

Fig. 7. Graph of Errors

shape does not include the person's hand as shown in Fig.8 (b). Lastly, the toy figure's texture was mapped on the extracted shape as shown in Fig.8 (c).

5 Conclusion and Future Work

In this paper, we showed the following results obtained with the wearable vision sensor. First, we analyzed and classified human manipulation for observation into four types and related them with acquirable visual information. Second, we proposed *dynamic space carving* for 3D shape extraction of a static object occluded by a dynamic object moving around the static object. Finally, we showed



(a) Captured Image

(b) Extracted Shape

(c) Texture Mapped Shape

Fig. 8. Toy Figure Data Set

that stereo depth map and silhouettes can be integrated in *vacant space*, and showed that our approach is effective by experiment.

Now, we are studying detection of 3D gazing object position, and camera control for capturing the gazing object appropriately. Therefore, we will integrate these methods with our approach of this paper in the future.

Acknowledgements

This paper is supported in part by the Ministry of Education, Culture, Sports, Science and Technology Grant No. 13224051.

References

[1] I, Napier.: The prehensile movements of the human hand. J.Bone and Joint Surgery, **38B**(4), (1956) 902–913. 129

[2] M, R, Cutkosky.: On Grasp Choice, Grasp Models, and the Design of Hands for Manufacturing Tasks. IEEE Trans. Robot. Automat., **5**(3), (1989) 269–279. 129

[3] W, N, Martin., J, K, Aggarwal.: Volumetric description of objects from multiple views. IEEE Transactions on Pattern Analysis and Machine Intelligence, **5**(2) (1987) 150–158. 130

[4] C, Tomasi., T, Kanade.: Shape and motion from image streams under orthography: a factorization method. Int'l Journal of Computer Vision, Vol. **9**(2) (1992) 137–154. 130

[5] H, Baker.: Three-dimensional modelling. Fifth International Joint Conference on Artificial Intelligence, (1977) 649–655. 130

[6] K, N, Kutulakos., S, M, Seitz.: A theory of shape by space carving. IEEE International Conference on Computer VisionC(1999) 307–314. 130, 131

[7] H, Hoppe,CT, DeRose.CT, Duchamp.CJ, McDonald.CW, Stuetzle.: Surface reconstruction from unorganized points. Computer Graphics (SIGGRAPH '92 Proceedings)Cvolume **26**C(July 1992) 71–78. 131

[8] A, Laurentini.: How far 3d shapes can be understood from 2d silhouettes IEEE Transactions on Pattern Analysis and Machine Intelligence, **17**(2), (1995) 188–195. 131

[9] C, J, Harris,, M, Stephens.: A combined corner and edge detector. In Proc. 4th Alvey Vision Conf.CManchesterC(1988) 147–151. 136

[10] P, J, Besl.CN, D, McKey.: A method for registration of 3-D shapes. IEEE Trans. Patt. Anal. Machine Intell.Cvol. **14**(2)C(1992) 239–256. 136

[11] Szymon, Rusinkiewicz.COlaf, Hall-Holt.C Marc, Levoy.: Real-Time 3D Model Acquisition. Transactions on Graphics (SIGGRAPH proceedings), (2002), 438–446. 131

[12] S, T, Barnard.: Stochastic approach to stereo vision. International Journal of Computer Vision, (1989) 17–32. 137

Location-Based Information Support System Using Multiple Cameras and LED Light Sources with the Compact Battery-Less Information Terminal (CoBIT)

Ikuko Shimizu Okatani[1] and Nishimura Takuichi[2]

[1] Tokyo University of Agriculture and Technology, Japan
[2] Cyber Assist Research Center
National Institute of Advanced Industrial Science and Technology, Japan

Abstract. To realize a location-based information support system, we propose an information support system using Compact Battery-less Information Terminals (CoBITs). This system consists of the information terminals CoBITs, and many environment system units. The environmental system unit plays an important role that to send particular information to respective users, and consists of multiple cameras for estimation of the position and orientation of the terminal, and LED light sources on the controllable pan-tilt heads for sending the optical beam information. For the application for information support in an event, a method for calibration of multiple cameras and light sources which is easy to use for inexperienced persons and to setup quickly is needed. To calibrate multiple cameras and LED light sources simultaneously, we employ a calibration technique for the multiple cameras based on the self-calibration method.

1 Introduction

In the near future, people will access information 'whenever, wherever, whoever,' and enjoy information services while moving around in the real world. In such the "ubiquitous"[1] and "pervasive"[2] computing environment, it is important to provide "context-aware"[3] information to a user in an easy operation.

Among enormous size of information, a lot of information is specific to a location will be embedded in that location. Therefore, it is important to realize location-aware information support that reflects users' situation and intention and provides appropriate information "here, now, for me" in an easy operation[4].

To realize such the information support, local communication will be important instead of global communication such as cellphones, PHS, or GPS [5] technology. Because a user's location and orientation can be used directly, local communication is suitable for sending and receiving information specific to a location.

Many information support systems which utilize the local communication have been proposed, such as C-MAP[6], Shopping Assistance[7], and

N. Sebe et al. (Eds.): HCI/ECCV 2004, LNCS 3058, pp. 142–152, 2004.

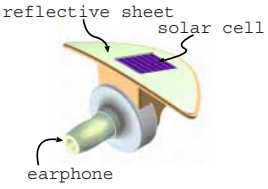


Fig. 1. The terminal, CoBIT which consists of a solar cell, an earphone, and a reflective sheet

Cyberguide[8]. They use high function terminals like PDA that have small displays and rich communication devices. But, it is preferred to be used by more easy operation for inexperienced users. In addition, in such the systems, a user have to see a monitor of the PDA, and it would prevent interaction with other people or environment.

To cope with these problems, we have proposed Compact Battery-less Information Terminal - CoBIT[9], which works using only the energy from the information carrier and the user. It is very compact, and has intuitive and interactive interface, no-latency, and no-battery.

In this paper, we propose a new location-aware information support system using CoBIT, multiple cameras and LED light sources. In this system, to send situation-dependent, location-dependent, and user-dependent information is sent to each user using LED light sources, the position and orientation of the CoBIT is estimated based on its observations by multiple cameras. By using multiple light sources in addition to multiple cameras at each location, user-dependent information can be sent to each user.

To realize this system, the relation between the observations by the cameras and the direction of the light source have to be known beforehand, i.e., cameras and light source have to be calibrated. Especially, for practical information support at event venues, such as a museum, at which unspecified number of people gather, simple calibration system which is easy to be used for inexperienced persons is needed.

In this paper, first the terminal, CoBIT, and the information support system using CoBIT is described, then a method for calibrating cameras and LED light sources is proposed, which is the key technique to realize such the information support system. The implemented prototype system and some experimental results are also shown.

2 The Compact Battery-Less Information Terminal, CoBIT

The CoBIT consists of a solar cell, an earphone, and a reflective sheet shown in Fig.1, and receives the light energy which contains information and converts it into electric current by utilizing this energy without battery. This energy

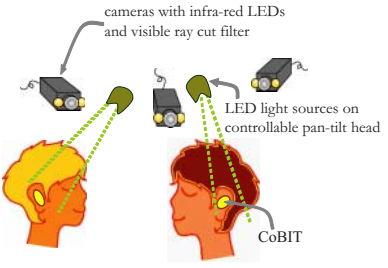


Fig. 2. The environmental system unit of the information support system using CoBIT

supplied from the environment activates an earphone in order to communicate with the user.

Many other types of CoBIT are being developed. For example, ID-CoBIT[10], equipped an IR ID tag and a liquid crystal shutter, can send its own decoded ID number of the terminal using infrared-red LED to identify the user of this CoBIT. In this device, an IR ID tag and a liquid crystal shutter are driven with no plug-in power source. The IR ID tag uploads a user's ID to the environmental system from several meters distance. The ID detector in the environmental system consists of an IR projection camera and an IR sensor.

3 Information Support System Using CoBIT

3.1 The Environmental System Unit

In our information support system using CoBIT shown in Fig. 2, the environmental system unit consists of multiple cameras and LED light sources on the controllable pan-tilt head. The position and orientation of the CoBIT are estimated based on its observations by multiple cameras, and the optical beams which are modulated based on the situation-dependent, location-dependent, and user-dependent information is sent to each user.

Because both multiple cameras and multiple light sources are equipped at each location, user-dependent information can be provided simultaneously to multiple users at the same location. For example, if this system is used in the museum, people gather in front of one painting can receive respective information. If a person is interestead in the artist of this work, the information related to that artist is provided. If the other person is a child, the information for children is provided. Futhermore, the interaction between a user and the system would enable to provide information that reflects users's condition, such as he/she is tired or not.

It is desirable to locate the cameras and LED light sources at the higher position than people's head position, for example, the top of the wall where the painting is displayed. It is because this configuration may prevent the blind area of the location.

Fig. 3. The cameras with infra-red LEDs and the visible ray cut filter located in the environment

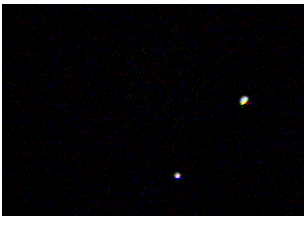


Fig. 4. The image captured by the camera located in the environment

3.2 Observation of the CoBIT by Cameras

The position of the terminal, CoBIT, is estimated from the observation of the terminal by multiple cameras in the environmental system unit. The cameras with infra-red LEDs are attached with the visible ray cut filter, as shown in Fig.3. On the other hand, the reflective sheet attached with the CoBIT includes many small-sized corner reflector cubes and reflects optical beam back in the direction of the light source. Therefore, a CoBIT is observed in a camera image as a glittering point, and is easily detected in the image. Figure 4 shows two CoBITs in the image observed by the camera with infra-red LEDs and the visible ray cut filter.

Once the cameras are calibrated, from the observations of a CoBIT by two or more cameras, its position can be estimated using its observed position in the image and cameras' extrinsic and intrinsic parameters. Therefore, by calibrating the intrinsic parameters (focal length etc.) and the extrinsic parameters (the positions and orientations) of cameras beforehand, the position of the terminal can be estimated using multiple images. Furthermore, to send information to the CoBIT, the relative position and orientation of the light sources on the pan-tilt head. So that, we calibrate the multiple cameras and the light sources simultaneously.

3.3 Sending Information to the CoBIT

The optical beams modulated by the sound information are emitted by the LED light source on the controllable pan-tilt head to each user's position estimated from the observations by multiple cameras.

To realize user-dependent information support, respective LED light sources send different optical beams to respective users by controlling the directions of the LED light sources. To control the directions of the LED light sources, the relation between the estimated user's position by cameras and the projecting direction of the optical beam by LEDs are to be calibrated each other.

To calibrate cameras and LED light sources, the images of the optical beams projected on a plane by the LED is captured by the cameras, as details are explained in the next section. The accuracy of the calibration is improved when the the beams projected by the LED are thinner, while it should be wider to provide the information to users robustly. To cope with this problem, we have two types of LED light sources on one pan-tiled head: One is used for sending information. The optical beam emitted by this LED light source is projected onto conic area whose vertical angle is about 10 degrees. Another is used for calibration whose vertical angle of this LED's projection area is about 5 degrees.

4 The Simultaneous Calibration of Cameras and LED Light Sources

In order to realize the user-dependent information support, it is necessary to know the relative position and orientation of each camera and LED light source. In other words, cameras and LED light sources should be calibrated beforehand. It is important to develop a convenient, simple and adaptive calibration system that takes not so long time to calibrate cameras and LEDs, especially when this system is used by inexperienced persons, for example, in shops or museums, or in some event venues. Moreover, it is desirable to cope with the unexpected movement of some of cameras.

Therefore, we simultaneously calibrate multiple cameras for estimation of the position and orientation of the terminal, and LED light sources for sending the optical beam modulated by the sound information to each user.

By calibrating multiple cameras and LED light sources simultaneously, the relative position and orientation of each camera and LED light source, parameters which characterize each camera, such as focal length, image center, etc., and the projecting direction of LED are to be known. The position and orientation of cameras and LEDs are called 'extrinsic parameters', and the focal length etc. of cameras and the direction of projection of LEDs are called 'intrinsic parameters'.

The projecting direction of the LED light source is controlled by the pan-tilt head. This LED system can be thought as a virtual camera by converting the pan-tilt angle to a virtual image coordinate. Then, the calibration methods for multiple cameras can be applied to our system consisting of multiple cameras and LED light sources.

As mentioned above, we use the images of the optical beams projected on a plane by the LED. The image of the optical beam projected by the LED for calibration is invisible, but they can be observed by the camera is shown in Fig.5. To calibrate the cameras and the LED light sources, many optical

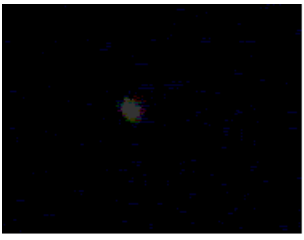


Fig. 5. The image of the optical beam projected on a plane is observed by cameras

beams are projected from the LEDs to various directions on planes located in the environment, and the centroid of the elliptic projected areas are used.

Many camera calibration methods have been proposed in the literature[11, 12, 13, 14]. In our case for calibrating cameras and light sources, we have to use the images of projected optical beams, i.e., we can know the direction of the light source and its image observed by cameras, and the 3D coordinates of reference points can not be known. Therefore, we employ self-calibration methods[13, 15] which need only the correspondences of the reference points between images, and don't need 3D coordinates of the points.

For each environmental system unit, multiple cameras and multiple LED light sources are located. Calibration is done for all cameras and each LED light source at each unit.

4.1 Relationship between the World Coordinate System and the Camera Coordinate System

Assume that there are N cameras in one environmental system unit. An LED system can be thought as a $N+1$-th camera by converting the pan-tilt angle (θ, ϕ) to a virtual image coordinate $(f\cos(\theta)/\tan(\phi), f\sin(\theta)/\tan(\phi))$ where f is the virtual focal length.

We assume that the position of the centroid of the projected area expressed in the world coordinate system is $[X \;\; Y \;\; Z]^\top$, its position expressed in the i-th camera coordinate system depends on the position and orientation of the i-th camera is $[X_i \;\; Y_i \;\; Z_i]^\top$, its observed position in the i-th image coordinate system is $[x_i \;\; y_i]^\top$.

The relation between $[X_i \;\; Y_i \;\; Z_i]^\top$ and $[x_i \;\; y_i]^\top$ can be expressed as

$$s \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = \boldsymbol{P}_i \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}, \tag{1}$$

where s is an arbitrary scale factor. The projection matrix $\boldsymbol{P}_i$ can be decomposed as $\boldsymbol{P}_i = \boldsymbol{K}_i\boldsymbol{T}_i$, where $\boldsymbol{K}_i$ is 3×3 camera intrinsic matrix of the i-th camera which consists of the intrinsic parameters such as the focal length etc. of the camera, and $\boldsymbol{T}_i$ is 3×4 camera extrinsic matrix of the i-th camera.

The matrix $\boldsymbol{T}_i$ is further decomposed using a rotation matrix $\boldsymbol{R}_i$ and a translation vector $\boldsymbol{t}_i$ as $\boldsymbol{T}_i = [\boldsymbol{R}_i|\boldsymbol{t}_i]$. The rotation matrix $\boldsymbol{R}_i$ of the i-th camera expresses the orientation of the i-th camera, and the translation vector $\boldsymbol{t}_i$ expresses the position of the i-th camera. By using $\boldsymbol{R}_i$ and $\boldsymbol{t}_i$, we have $[X_i \;\; Y_i \;\; Z_i]^\top = R_i[X \;\; Y \;\; Z]^\top + \boldsymbol{t}_i$.

By substituting these relations, the right-hand side of Eqn. (1) is rewritten as

$$\boldsymbol{P}_i \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix} = \boldsymbol{K}_i \boldsymbol{T}_i \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix} = K_i \left[R_i \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \boldsymbol{t}_i \right] = \boldsymbol{K}_i \begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix}. \tag{2}$$

On the other hand, the matrix $\boldsymbol{K}_i$ is dependent on the camera model. We use the perspective camera model expressed as

$$\boldsymbol{K}_i = \begin{bmatrix} f_i & s_i f_i & u_i \\ 0 & \alpha_i f_i & v_i \\ 0 & 0 & 1 \end{bmatrix},$$

where f_i is the focal length, α_i is the aspect ratio, s_i is the skew, and the (u_i, v_i) is the image center of the i-th camera. Then, the right-hand side of Eqn. (2) is rewritten as

$$x_i = \frac{f_i X_i + s_i f_i Y_i}{Z_i} + u_i, \quad y_i = \frac{\alpha_i f_i Y_i}{Z_i} + v_i. \tag{3}$$

Furthermore, the radial lens distortion of the cameras should be considered in this system. Because the cameras located in the environmental system unit are desirable to have wide fields of views to reduce the blind area, the effect of the lens distortion is significant.

To consider the radial lens distortion, the relation between the observed coordinate (x'_i, y'_i) in the image and the undistorted image coordinate (x_i, y_i) is expressed as

$$x'_i = x_i + (x_i - u_i)[k_{1i} r^2 + k_{2i} r^4], \quad y'_i = y_i + (y_i - v_i)[k_{1i} r^2 + k_{2i} r^4], \tag{4}$$

where k_{1i} is the first order radial lens distortion coefficient, k_{2i} is the second order radial lens distortion coefficient, and r^2 is equal to $(x_i - u_i)^2 + (y_i - v_i)^2$.

4.2 Calibration of Multiple Cameras and LEDs

To know the camera intrinsic and extrinsic parameters beforehand, the cameras should be calibrated. In this research, we apply a multiple camera calibration method based on the self-calibration[15].

In this method, by using the correspondences of the reference points between multiple cameras, the intrinsic and extrinsic camera parameters of multiple cameras, i.e., $\boldsymbol{P}_i$ in Eqn.(1) are estimated simultaneously based on the bundle adjustment. The advantage of this method is to stably estimate the initial value of the nonlinear minimization by switching the camera model from the simple model, i.e., the orthographic model, to the complex model, i.e., the perspective projection step by step.

Though this method does not use the 3D coordinates of the reference points, to resolve the scale ambiguity, we use the 3D coordinates of the camera positions.

4.3 Estimation of the Position of the Terminal

By using the intrinsic and extrinsic parameters of multiple cameras estimated by camera calibration mentioned in the previous section, the 3D position of the terminal is estimated by the image coordinates observed by the multiple cameras as followings.

First, from the distorted image coordinate $[x'_i \quad y'_i]^\top$ of the reference point observed by the i-th camera, the undistorted image coordinate $[x_i \quad y_i]^\top$ is estimated from Eqn.(4). Then, the 3D coordinate $[X \quad Y \quad Z]^\top$ in the world coordinate is estimated using Eqn.(3) by minimizing the next criterion

$$
\begin{aligned}
J &= \textstyle\sum_{i=1}^{N} \varepsilon_i [(x_i - \hat{x}(\boldsymbol{K}_i, \boldsymbol{T}_i, X, Y, Z))^2 + (y_i - \hat{y}(\boldsymbol{K}_i, \boldsymbol{T}_i, X, Y, Z))^2] \\
&= \textstyle\sum_{i=1}^{N} \varepsilon_i [\left(x_i - \left(\frac{f_i X_i(\boldsymbol{T}_i, X, Y, Z) + s_i f_i Y_i(\boldsymbol{T}_i, X, Y, Z)}{Z_i(\boldsymbol{T}_i, X, Y, Z)} + u_i \right) \right)^2 \\
&\qquad + \left(y_i - \left(\frac{\alpha_i f_i Y_i(\boldsymbol{T}_i, X, Y, Z)}{Z_i(\boldsymbol{T}_i, X, Y, Z)} + v_i \right) \right)^2],
\end{aligned}
$$

where ε_i is equal to 1 if it is observed from the i-th camera, otherwise it is equal to 0.

5 Experimental Result

To test our method, we implemented a prototype system consists of two cameras and one LED light source.

The used reference points were projected from the light source to 63 directions. From the correspondences of the projection angle and the centroid of the projected area of the optical beam projected on a plane in each image observed by each camera, the camera parameters were calculated by above mentioned method. The scale and the world coordinate system was adjusted by the 3D coordinate of the camera positions.

The results are shown in Fig.6. The 3D positions of each camera, i.e., the calibrated extrinsic parameters, and those of the reference points calculated using calibrated intrinsic and extrinsic parameters of each camera and LED light source are shown.

Using these parameters, the position of the CoBIT was calculated by observations of two cameras and the information was projected from the light source.

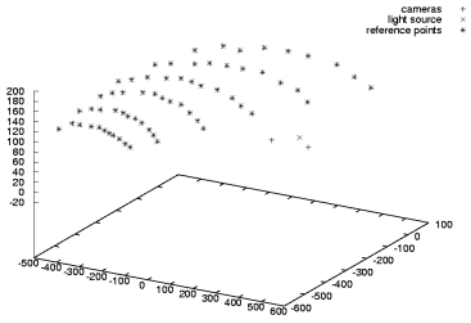


Fig. 6. The 3D positions of each camera and those of the reference points calculated using calibrated intrinsic and extrinsic parameters of each camera

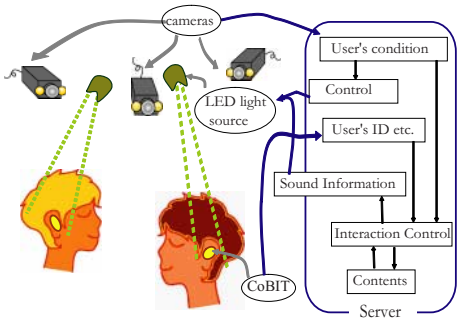


Fig. 7. The future information support system

The information can be heard using the CoBIT. Though more verification is needed, we ascertain that our calibration method is useful for this information support system.

6 Future System

The overview of the system is shown in Fig.7. To complete overall system, the important task is to generate situation-dependent, location-dependent, and user-dependent information by using ID-CoBIT. And, the control of handover between the environmental system units is also important.

To generate situation-dependent information, the records of the users' position are very useful. We have been done some experiments to track multiple users simultaneously in one location.

7 Conclusion

In this paper, a location-aware information support system using multiple cameras and LED light sources on the controllable pan-tilt heads with Compact

Battery-less Information Terminals (CoBITs) is proposed. In this system, the cameras are used for estimation of the position and orientation of the terminal, while the LED light sources are used for sending the optical beam information. To calibrate multiple cameras and LED light sources simultaneously, we employ a calibration technique for the multiple cameras based on the self-calibration method. Our system enable that user-dependent information be provided simultaneously to multiple persons at the same location because both multiple cameras and multiple light sources are equipped at each location.

References

[1] Weiser, M.: Some Computer Science Issues in Ubiquitous Computing. Communications of the ACM **36** (1993) 74–84 142

[2] Satyanarayanan, M.: Pervasive Computing : Vision and Challenges. IEEE Personal Communications (2001) 10–17 142

[3] Schilit, B., Adams, N., Want, R.: Context-Aware Computing Applications. In: Proc. of IEEE Workshop on Mobile Computing Systems and Applications. (2001) 85–90 142

[4] Nakashima, H., Hasida, K.: Location-based communication infrastructure for situated human support. In: Proc. of SCI. (2001) 47–51 142

[5] Loomis, J. M., et al.: Personal Guidance System for the Visually Impaired. In: Proc. of First Annual International ACM/SIGCAPH Conf. on Assistive Technologies. (1992) 91–102 142

[6] Etani, Y. S. T., Fels, S., Simonet, N., Kobayashi, K., Mase, K.: C-MAP: Building a context-aware mobile assistant for exhibition tours. In: Proc. of The First Kyoto Meeting on Social Interaction and Community Ware. (1998) 142

[7] Bohnenberger, T., Jameson, A., Kruger, A., Butz, A.: Location-Aware Shopping Assistance: Evaluation of a Decision-Theoretic Approach. In: Proc. of the Fourth International Symposium on Human Computer Interaction with Mobile Devices. (2002) 142

[8] Anowd, G. D., Atkeson, C. G., Hong, J., Long, S., Kooper, R., Pinkerton, M.: Cyberguide: A mobile context-aware tour guide. Wireless Networks **3** (1997) 421–433 143

[9] Nishimura, T., Itoh, H., Yamamoto, Y., Nakashima, H.: A Compact Battery-less Information Terminal (CoBIT) for Location-based Support Systems. In: Proc. of SPIE. Number 4863B-12 (2002) 143

[10] Nakamura, Y., Nishimura, T., Itoh, H., Nakashima, H.: An ID Output Type Terminal: ID-CoBIT for Location Based Personal Information Support(In Japanese). Information Processing Society of Japan SIG Notes **ITS-11** (2002) 7–12 144

[11] Tsai, R.: A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses. IEEE Journal of Robotics and Automation **RA-3** (1987) 323–344 147

[12] Quan, L.: Self-Calibration of an Affine Camera from Multiple Views. International Journal of Computer Vision **19** (1996) 93–105 147

[13] Pollefeys, M., Koch, R., Gool, L. J. V.: Self-Calibration and Metric Reconstruction Inspite of Varying and Unknown Intrinsic Camera Parameters. International Journal of Computer Vision **32** (1999) 7–25 147

[14] Zhang, Z., Loop, C.: Estimating the Fundamental Matrix by Transforming Image Points in Projective Space. Computer Vision and Image Understanding **82** (2001) 174–180 147

[15] Deguchi, K., Okatani, T.: Calibration of multi-view cameras for 3d scene understanding(in japanese). Information Processing Society of Japan SIG Notes **CVIM-131** (2002) 1–8 147, 148

Djinn: Interaction Framework for Home Environment Using Speech and Vision

Jan Kleindienst, Tomáš Macek, Ladislav Serédi, and Jan Šedivý

IBM Česká republika, Voice Technologies and Systems
The Park, V Parku 2294/4, 148 00 Praha 4 Chodov, Česká republika
{jankle,tomas_macek,ladislav_seredi,jan_sedivy}@cz.ibm.com

Abstract. In this paper we describe an interaction framework that uses speech recognition and computer vision to model new generation of interfaces in the residential environment. We outline the blueprints of the architecture and describe the main building blocks. We show a concrete prototype platform where this novel architecture has been deployed and will be tested at the user field trials. EC co-funds this work as part of HomeTalk IST-2001-33507 project.

1 Motivation and Goals

The world around us is changing. Several decades ago computers have entered our lives and become one of the key driving forces of the change. The prospect they offer is to make us more creative, more productive, and more free. Historically, we have learned to interact with our computers via graphical displays using keyboards and mice. We have grown so accustomed to it that for the youngsters this seems almost natural.

However, the traditional concept of computer application is now rapidly expanding. Applications started to reach out of PC boxes and proliferate to our settings to ease everyday tasks. In not very distant future, the applications will blend in and become part of our environment. We users will be living in applications, interacting with them from inside. That is a true paradigm shift. This revolution has already started in the automotive industry. The car will soon become such an application. Homes and offices will follow next.

This new interaction model needs new interaction means. Mouse and keyboards, being part of the old PC-bound application scheme, will no longer suffice. We need new ways of engineering man-machine dialogs and we need new input and output interfaces. Some promising technologies -- like speech recognition and computer vision -- are already there to help.

Obviously, this paradigm shift will not happen overnight. It will need several technological waves sweeping across the research labs prototypes and industry solutions to create something elementary useful.

Our goal is to offer small contribution to this new emerging wave of user interface technologies. We describe a research framework called Djinn that harnesses speech and vision to model new ways of interaction in the residential environment.

N. Sebe et al. (Eds.): HCI/ECCV 2004, LNCS 3058, pp. 153-164, 2004.

Coming from speech recognition arena, with major background in architecting multi-modal applications [5,6,7,8,9], and basic knowledge of computer vision technology, we wanted to exercise our experience in a dramatically new type of environment. We intended to design an open and flexible framework that would allow to model novel interaction patterns in home environment based on speech and vision. We did not want to develop the user interface technologies, we wanted to use them [1,2] and combine them into a new "multi-modal" framework. Being on the common project with a white appliance manufacturer that provides ovens, washing machines, and refrigerators with remote control access our initial focus naturally drifted towards the kitchen setting. It has certainly turned out to be a good choice given the extent of UI challenges we are encountering along the way.

The above motivation standpoint plus the fact that the system will run in real-life conditions differentiate the focus of our architecture with respect to the existing systems in the arena of SmartHomes [10], agents [11], specific interaction gadgets [12,13], and novel UI approaches [14]. In comparison to state-of-the-art context acquisition and modeling systems [15,16], Djinn's added value is in treating not only context gathering but also multi-modal interactions as first-class entities.

2 Design Requirements and Approach

First we overview the design requirements which govern the development of the Djinn architecture and its primary goal to support specific user interaction patterns. Then we outline the approach and consequences it bears on the actual design and implementation.

Extend Multi-modality. Strech the concept of multi-modal applications to incorporate support for modeling environment. Build on multimodality's great advantage to provide application developers with a scalable blend of input and output channels that may accommodate any user, device, and platform – based on the circumstances, hardware capabilities, and environment.

Be Open & Flexible. Provide a general framework for modeling interactions in home environment. Make the framework open and flexible to allow for future extensions and parameterization. Base the design and implementation on the existing standards (XML, VoiceXML, Macromedia Flash, etc.)

Modularize. Construct the framework from reusable components as building blocks. Explicitly provide support for distribution to allow various components of the architecture to run on different machines. This is especially important to enable efficient processing to camera and other sensory input.

Provide General Distant Access. Provide an aggregated remote access to the devices and appliances. This will be useful for remote monitoring. It will also provide an alternative to people with special needs (elderly and disabled).

Use Speech. Use speech as main part of the user interface. Thanks to more than four decades of research in speech recognition technology we have now possibility to run speech recognition on small devices like PDAs with reasonable recognition rates.

Use Vision. Use the vision for providing cues about the environment and the user actions. Cameras are excellent sensors having broad information bandwidth. Let them provide information on subjects and related objects and their interactions.

Bind by Eventing. Connect the components using synchronous and asynchronous eventing. Events provide good abstractions over various pieces of information that would be floating around system and can be easily transferred over the network. In next steps of the architecture progress, they would allow to deploy various kinds of sensors to plug in into the framework. The more human-like the interaction should become, the more sensors we need to employ to acquire more context.

So with the respect to the requirements, we will now spend a few words on the approach chosen. Clearly, designing system like this is not a simple task. The classic "waterfall" approach of software architecture theory -- i.e. design and develop, then integrate and test, and you are done – obviously cannot be applied. Instead, we used a mixture of "evolution" and "user-centric" design methodologies. Starting with a very simple end-to-end system, we gradually "grow" it over time based on the user feedback and expected impact of to-be-added features. On regular intervals we try the system in our research laboratory equipped with several camera-mounted white appliances. The outcomes of these sessions constitute important input for continual design and implementation work.

We believe that one of the important design "tools" in the development process was the existence of a scenario. We put the scenario together at the beginning of the actual architectural work to provide basic development directions and to have means to ground our work. For illustration, here is a snippet of the scenario (the central character in the scenario is Mr. N.):

...When Mr. N. approaches the oven with the ready-to-cook pizza, a camera-mounted oven detects his face and displays the recommended cooking parameters for the pizza on the oven display. At the same time, it synchronizes the display of a multi-modal PDA assistant held by his mother-in-law. Mr. N. can thus adjust the values using either turning the oven knobs or the multi-modal oven interface on the PDA.

Peeking at the PDA display, his mother-in-law argues that the recommended temperature is too low and the cooking time too short. Mr. N. shrugging in defeat suggests she alter the values. She does so by speaking to the PDA ("Set cooking time to forty-five minutes. Increase the temperature to 200 degrees.") and encouraged, she even tries to fool the wizard with "This should be ready by 1 pm, so set the start time accordingly," astonished that the command actually works as the oven displays the calculated start time.

From her chair the old lady demands Mr.N. to show her how to start the oven remotely from the PDA. As Mr.N. moves out of the oven direct interaction zone, the camera-mounted oven detects that. Based on this cue, Djinn notifies him that if he

wants to start the cooking process he must press the START button – either on the oven panel or on the PDA…

To make the development more manageable, we partitioned the problem space into several areas and identified the components that cover them. As the primary integration points, we defined contracts among component interfaces. Three major components we describe in the rest of the paper.

The key component is the Djinn runtime (next section) – a hub which processes information from all user-bound, sensor-bound, and appliance–bound components. The other component we introduce is a visual-processing component called Visionary. Another major component that we describe is MACI, a multi-modal component for client interactions.

3 The Djinn Runtime

The Djinn runtime is the central component of the backend architecture – it collects context from various information sources in the environment, such as cameras, sensors, devices, appliances, etc., and provides means to control them. Specifically, the visual cues supplied by the camera subsystem allow the applications to use environment context and thus enhance user interactions. As we mentioned above, such context-sensitive applications can exhibit improved features in terms of presenting the information to the user, resolving disambiguities ("e.g. start this") and pro-actively solving tasks without user's explicit commands. This is the necessary prerequisite to enhance "intelligent" behavior of this and similar systems in cars, homes and offices.

Written in Java, the Djinn runtime takes advantage of the language support for eventing and conceptual abstractions. It builds on subscribers/listener pattern to cater for delivering proper pieces of information from producers to consumers.

Djinn as Event Hub. The Djinn runtime provides several service interfaces. We now describe the major ones:

- *HTTP service* – for sending and receiving HTTP traffic and also for pushing asynchronous information to the HTTP clients
- *Camera/sensor service* – for receiving scene-related camera events (e.g., detecting the proximity of the user) and controlling output devices (e.g. the laser pointer)
- *Appliance services* – for interaction with various appliances, e.g. white appliances and sensors

Djinn applications, called Djinnlets, subscribe to specific events on each service interface and the Djinn runtime keeps them up-to-date with inbound messages generated on the service interfaces. All inbound traffic (from user clients, appliances, cameras, and sensors) is turned into events and dispatched to respective services. The event processing infrastructure is scalable and open enough to add events (and services) quite easily. All events are stored in the searchable context history which is

an important prerequisite for implementing advanced dialog and appliance management strategies as well as supporting automatic adaptation for a given user.

Djinn as State Manager. Good programming framework has to provide useful abstractions. The kind of applications we are dealing with need to keep track of the state of the appliances and devices they interact with. For example, they need to know whether an oven is currently in cooking mode, programming mode, or idle. Djinn makes those abstractions available to Djinnlet applications by modeling the appliance state. The underlying Djinn runtime ensures that these virtual appliances are synchronized with real appliances (using the specific power-line protocol). There are several advantages to such an appliance modeling:

- it is general enough to be used by many Djinn applications
- the applications do not have to make interpretation of the low-level power-line event traffic (even though they may do so if needed)
- virtual appliances can emulate real appliances even if the latter are not available, such as in case of tuning and debugging
- virtual appliances can capture broader contextual information (sensory, temporal) than they real counterparts, e.g. the distance of the user from the appliance, the interaction history, (e.g. previous recipes, user interaction sequences, etc.)

Capturing and interpreting as much context as possible helps cope with non-obvious pitfalls of user interactions. This includes modeling not only the state of the appliance but also the "channels" of user interactions. We explain what we mean using the case of a camera-mounted oven, but the concept applies to any device in general.

The oven operates in three basic states: in the idle mode, the oven display shows just real-time clock. In the programming mode, the display shows info about current recipe (26 built-in recipes), temperature, cooking time, duration etc. By pressing a button or using a speech command, the oven enters the cooking mode. In our system, the user can switch the oven into the programming mode by three different ways:

1. Turning a knob on the oven.
2. Pressing a button on the PDA (as a specific case of HomeTalk control device).
3. Stepping in front of the oven in which case the "face-in" camera event automatically switches the oven into programming mode, displaying last recipe found in the context history.

Suppose now Option 3. If the oven-facing user does not engage in the interaction and leaves the oven proximity, the "face-out" camera event dims the oven display back to the idle mode after a short time. (This behavior is similar to a clerk smiling and making an "eye-contact" but then relaxing again when the customer walks-by and shows no real interest.). If the user engages in the interaction with the oven either directly operating the control panel or using the PDA-based oven controls, the "face-out" event causing automatic fallback to the idle mode should be suppressed even though the user leaves the oven interaction zone. If not, the system behavior would be considered unnatural.

Similarly, suppose a user operating from a distance uses its PDA controls to wake up the oven from an idle state enter a particular recipe (Option 2). When (s)he now enters the proximity of the oven, the "face-on" event should be basically ignored because the user in already engaged in the interaction. (Otherwise, the oven would with the "face-in" event overwrite the current recipe entered from PDA with the previous one retrieved from the history.) The same applies to "face-out" event. One could argue that such a behavior should change if there passes enough time between user operating the PDA and entering the interaction zone in such a case.

From the example above, it is clear the visual cues constitute an important part of the system. Their acquisition and interpretation is crucial to make the system act naturally. In our system we currently work with the concept of appliance interaction zones, where we try to capture and model what is happening in front of camera-mounted appliance. Through the Visionary component (Section Visionary), we can detect objects moving in front of the appliance, and can distinguish subject's faces from background moving objects. It is also possible to induce the relative distance of the subject from the appliance. It is obvious that additional more "intelligent" cues would help improve the user satisfaction of such an interaction system and increase robustness of existing applications, e.g.:

- *Detecting speaker intent* – open-microphone solution is very hard in noisy environment like kitchen. Visual cues from lip-tracking camera with help to indicate the beginning of speech, and thus the user will not need to use e.g. the push-to-speak button.
- *Speaker identity* – the system can detect the change of the speakers in front of the appliance and use this information to personalize application setting (project a different image, utilize the user context collected in previous interactions etc.)
- *Speaker attention* – the system can detect whether the user looks directly to the appliance, or to the projected menu, or if s/he is looking away. Such cues can be used as input for the multi-modal dialog manager to provide better feedback
- *Improve speech recognition performance* – under degrading acoustic conditions the recognition performance decreases significantly. The additional "lip-reading" information from the video channel will help reduce the error rate under unfavorable conditions (kitchen noise)
- *Multi-modal gestures* – the projector camera will detect user's pointing to a particular images, buttons, and selections on the projected menus. It may also determine the use actions on the appliance itself such as (opening /closing the oven doors)

Djinn as Event Aggregator. In process of development of the Djinn architecture, we soon found that some reusable logic is needed to combine different events. The Djinn runtime thus provides special objects called Djinn Aggregators for basic aggregation of low-level events to produce some higher semantic information. For example, the application can subscribe to an aggregator that emits an event in case a two specific events happen within certain time window (called temporal AND aggregator). E.g., if a door-sensor event and an door-camera "face-in" event happens simultaneously, the specific aggregator sends out an event to the application to indicate that someone is

entering the door (in contrast to leaving the door). Temporal OR aggregator is an example of an object that emits an output event if any predefined input event happens within a specified time window.

4 Visionary

Visionary is the component that provides high level camera services to the overall system. We separated visual processing to a separate subsystem. This is the decision which has been made based on the type of the applications we are dealing with. The examples of the applications are spotting the objects, humans, gestures, movement and modifying dialog flow based on visual clues. The application logic needs to know when certain event happens (image recognized, brightness changed) and parameters of the event (magnitude of the change, ID of the recognized face, distance of the moving object). The video stream itself is not of an interest. In some cases an image or a short video sequence is needed, typically taken shortly before and/or after the moment of the event. For example image of the intruder moving in the house. Instead of sending the images to the main logic of the application (where it is not needed), it is more convenient to make it available to all other components of the system. Then it can be grabbed directly by output subsystem.

We use single socket connection to transfer events and control messages. Thus, the Visionary could either run on the same physical machine as other components of the system or it can run on a dedicated computer to which cameras or other video stream resources are typically connected. It simplifies also overall development process. The same Visionary can be used with test bed implemented on PC as well as latter when the main application is ported to home gateway.

We decided to use XML format of data exchanged between Visionary and outside world. Using XML brings small additional processing overhead. But it simplifies future integration of Visionary to other systems. It removes dependency on binary coding (big endian, little endian). Efficient XML parsers are currently available.

The socket connection is used for transferring short events and its parameters. To transfer the images or video sequences we use a different mechanism. Visionary shares the disk space with a Web server. The Web server provides a mechanism to access the images by other components of the overall system.

The video stream images are recorded by Visionary to the circular image buffer. The images are kept in the buffer together with their time stamps. Buffer is continuously updated so the oldest images are overwritten by new ones. The application can, therefore, request an image taken at the particular time, presumed that the time is not too far in the history. The specific images (or video sequences) which are of the particular interest of the application can be stored by Visionary on request, independently on image buffer mechanism.

4.1 Communication Mechanism

Functions of Visionary can be either subscribed or requested by a message sent to the Visionary.

Examples of the requested service can be e.g. a request to take a picture or a request to return position of the laser pointer. An example of the message which has to be sent to Visionary and its response is shown below.

When sending the message:

```
<?xml version="1.0" ?>
<sensor_request version="0.1" originator="gateway_G1"
handle="13">
  <get target="camera_A">
    <screenshot timestamp="8:11:59" />
  </get>
</sensor_request>
```

Visionary sends in response the message:

```
<?xml version=\"1.0\" ?>
<sensor_event version="0.1" originator="camera_A"
timestamp="1:20:30" handle="13">
  <screenshot imgsrc="http://cs/visionar/001.jpg"/>
</sensor_event>
```

where *timestamp* is the time when the picture has been recorded (it can be different from requested time if the recorded image with exactly the time is not available). The *handle* is the number which is the same as the handle number in request (it couples the request and response). The *imagesrc* attribute is the URI of the image. It can be then requested separately by main logic of the application or bypassed further to the output subsystem.

Some of the events are initiated by the Visionary itself. This typically means any reporting about something happening in the video stream. For this type of operations we use a subscription mechanism. The logic of the application informs Visionary that it needs certain type of events by registering via a subscription message. For example, the application can subscribe to messages triggered by any movement on the scene. The *value* parameter in the subscription message controls application registration/deregistration. Further parameters of the algorithm (threshold of sensitivity, frequency of sending the events…) can be indicated in the subscription message.

We implemented Visionary in Visual C++. OpenCV [4] has been used to implement vision-related algorithms. XML document parsing and composition have been done using TinyXML parser. In spite of the fact that it is called "Tiny", it satisfied the needs of the application, it was small and easy to integrate with our code. We used Apache web server to serve the images. OpenCV, TinyXML and Apache are freely available under their licenses.

Currently we implemented the following image processing algorithms:

- *Movement detection* – Visionary calculates change in subsequent images. If it is larger than a threshold, it raises an event.
- *Human face detection* – Visionary detects presence of human face in front of the camera. It reports if the face appears, disappears on changes the size.
- *Laser pointer detection* – Algorithm detects position of laser pointer spot on a picture and reports its coordinates.

As one could expect, the whole system and its particular algorithms deals with many parameters which modify its functionality. We use structured XML configuration file to specify all of them at the moment of startup. Then most of the parameters can be also set by a special request sent to Visionary.

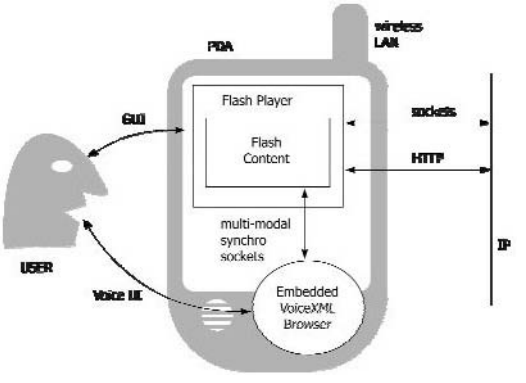


Fig. 1. Multi-Modal Client Architecture

Fig. 2. "Cooking Wizard" screenshot

5 MACI - Multimodal Appliance Control Interface

Appliances and devices integrated into the Djinn framework can be controlled not only directly in their interaction zones (assisted with the Visionary discussed above). They can be also controlled remotely by the clients connected to the Djinn runtime. These clients are either standard devices – i.e. a PC equipped with www browser, a telephone or mobile phone - or custom-made controllers with specific functionality and user interface.

To experiment with innovative HCI techniques we choose to implement a mobile client running on an off-the-shelf PDA, namely HP iPaq equipped with PocketPC 2002 operating system. This multi-modal mobile client combines the capabilities of IBM speech recognition technology with strength of a well-established GUI development environment. Our multi-modal framework, named as MACI (Multi-Modal Appliance Control Interface) reuses existing technologies such as the Macromedia Flash format for designing a visually appealing media-rich GUI and VoiceXML, industry-standard for the speech interactions. The binding between the two is provided by a XML-based synchronization protocol. See Fig. 1.

All components of our multi-modal framework are connected to the Djinn runtime trough wireless LAN. The connection includes a socket channel for delivering notifications and commands and HTTP for delivering binary data (such as images, Flash modules, etc.).

The appliances control interface consists of the following layers written in ActionScript (Macromedia's JavaScript derivate):

1. Djinn client; receiving events from the server and sending back commands through a persistent socket connection as well as receiving data through a HTTP channel
2. Voice server client, communicating with the embedded VoiceXML browser, i.e. sending in VoiceXML markup and receiving recognized phrases etc.
3. Macromedia Flash graphic front-end; capable to download and display GUI of any Djinn-compatible device packaged as a Flash module.

Djinn users usually interact with appliances or actuators through these downloadable device GUI modules which makes the system flexible and easily expandable.

An example of the Flash GUI can be seen on Fig. 2.

The graphic front-end consists of several parts:

1. The speech recognition interface – provides necessary feedback for speech recognition (level meter, last recognized phrase, etc.),
2. The Djinn connection indicator,
3. Status line informing the user about the actions carried out (i.e. “Cooking in progress”, “Washing process finished”, “Error occurred while cooking”, etc),
4. Appliance-specific control interface. Currently a Merloni oven and washing machine are supported. To switch between individual interfaces tap the tab with a respective icon at the lower edge of the screen.
5. Tabbed interface to switch between appliance control interfaces.

The Djinn-compatible device controller service described above is more than just an expensive universal remote controller. It is an integrated system which is both modular and extensible. The interface pages provide the user with a visual feedback about the current state of appliance. Emergency states and alarms from all appliances (i.e. washing machine spilling water) or sensors (i.e. intrusion detector, smoke detector) are triggered regardless of the actual state of the GUI front end. The cause of the alarm is usually given at the status line, with additional info is shown on the respective appliance/sensor interface page.

6 Use Case: Hometalk

The Djinn framework described above has been demonstrated as part of the HomeTalk project (IST-2001-33507). HomeTalk goal is to define and implement voice-enabled networking platform for home devices. The heart of HomeTalk architecture is the residential gateway, which is StrongARM 209 MHz Linux machine designed for 24/7 home use (low-consumption, no fan, no hard-drive, flash memory). The Djinn runtime is designed to run on the residential gateway (possibly as an OSGI bundle). The residential gateway also hosts VoiceXML browser to allow remote telephone access to HomeTalk devices. Last but not least, the residential gateway also hosts platform services that allow access remote devices and appliances over the LON power-line protocol. Due to high CPU requirements, Visionary runs on a dedicated PC connected with the Djinn runtime over Ethernet. The MACI runs on HP iPaq.

MACI, as a multi-modal component collects user input and interacts with the Djinn runtime.

Several HomeTalk services have been developed in the course of the project, such as “Cooking wizard” or “Alarm notifier”. The technology developed will be tested and evaluated this year at the field trials in Madrid and Athens. Part of the trials will include private residential households of technology enthusiasts as well as elderly and disabled users.

7 Summary

This paper describes a research framework called Djinn that uses aural and visual interfaces to provide access to devices and appliances in the residential environment. We did not intend to develop the user interface technologies; we integrated them into the framework as reusable components. Both those interface components - Visionary an MACI - were introduced in this paper. Also described is the runtime of the Djinn framework, a hub which processes and stores all data coming from client devices, appliances, cameras, and other possible sensors.

We have grown the framework based on the initial set of design requirements and the scenario as well as the feedback from our laboratory testing. As part of the HomeTalk project, the Djinn framework is going to be evaluated in field user trials.

As an acknowledgement, the authors would like to thank all HomeTalk (http://www.hometalk.org) partners - IBM, MERLONNI, WRAP, ANCO, TID, TEMAGON, INACCESS NETWORKS - for their valuable contributions to the work presented in this paper.

References

[1] VoiceXML 2.0, W3C, http://www.w3.org/TR/2001/WD-voicexml20-20011023, W3C Working Draft, (2001)

[2] Macromedia Flash, http://www.macromedia.com/software/flash/

[3] Multimodal Requirements for Voice Markup Languages, W3C, http://www.w3.org/TR/multimodal-reqs, W3C Working Draft, June 2000

[4] Intel, Open Source Computer Vision Library, http://sourceforge.net/projects/opencvlibrary/

[5] Ramaswamy, G., Kleindienst, J., Cofman, D., Gopalakrishnan, P., Neti, C. :A pervasive conversational interface for information interaction. Eurospeech 99, Budapest, Hungary, (1999)

[6] Fischer, V., Gunther, C., Ivanecky, J., Kunzmann, S., Sedivy, J., Ures, L.: Multi-modal interface for embedded devices. devices. Submitted to Elektronische Sprachsignalverarbeitung ESSV,TSc Dresden, Germany (2002)

[7] Demesticha, V., Gergic, J., Kleindienst, J., Mast, M., Polymenakos, L., Schulz, H., Seredi, L.: Aspects of design and implementation of multi-channel and multi-modal information system. ICSM2001, Italy, (2001)

[8] Kleindienst, J., Seredi, L., Kapanen, P., Bergman, J.: CATCH-2004 Multi-Modal Browser: Overview Description with Usability Analysis, IEEE Fourth, International Conference on Multimodal Interfaces, Pittsburg, USA (2002)

[9] Kleindienst, J., Seredi, L., Kapanen, P., Bergman J.: Loosely-coupled approach towards multi-modal browsing, Special Issue "Multimodality: a step towards universal access" of the Springer International Journal, Universal Access in the Information Society (2003)
[10] Smart Homes: a user perspective, 19th International Symposium on Human Factors in Telecommunication, Berlin, Germany, 1-4 December 2003
[11] Valerie Issarny: Offering a Consumer-Oriented Ambient Intelligence Environment, ERCIM News No.47, October 2001
[12] Jukka Vanhala: A Flood of Intelligence - the Living Room Project, ERCIM News No.47, October 2001
[13] Friedemann Mattern: Ubiquitous Computing Infrastructures, ERCIM News No.47, October 2001
[14] Mc Alester, D. , Capraro M.: Skip Intro: Flash Usability and Interface Design, New Riders Publishing (2002)
[15] MIT Oxygen Project, http://oxygen.lcs.mit.edu/Overview.html
[16] The Context Toolkit, http://www.cc.gatech.edu/fce/contexttoolkit/

A Novel Wearable System for Capturing User View Images

Hirotake Yamazoe[1,2], Akira Utsumi[1],
Nobuji Tetsutani[1], and Masahiko Yachida[2]

[1] ATR Media Information Science Laboratories
2-2-2 Hikaridai Seika-cho Soraku-gun, Kyoto 619-0288, Japan
[2] Graduate School of Engineering Science, Osaka University
1-3 Machikaneyama-cho Toyonaka-shi, Osaka 560-8531, Japan

Abstract. In this paper, we propose a body attached system to capture the experience of a person in sequence as audio/visual information. The proposed system consists of two cameras (one IR (infra-red) camera and one wide-angle color camera) and a microphone. The IR camera image is used for capturing the user's head motions. The wide-angle color camera is used for capturing frontal view images, and an image region approximately corresponding to the users' view is selected according to the estimated human head motions. The selected image and head motion data are stored in a storage device with audio data. This system can overcome the disadvantages of systems using head-mounted cameras in terms of the ease in putting on/taking off the device and its less obtrusive visual impact on third persons. Using the proposed system, we can record audio data, images in the user's view and head gestures (nodding, shaking, etc.) simultaneously. These data contain significant information for recording/analyzing human activities and can be used in wider application domains (such as a digital diary or interaction analysis). Experimental results show the effectiveness of the proposed system.

1 Introduction

The current down-sizing of computers and sensory devices will allow humans to wear these devices in a manner similar to clothes. This concept is known as 'wearable computing' [1, 2]. One major direction of wearable computing research is to smartly assist humans in daily life everywhere a user chooses to go. To achieve these capabilities, the system must recognize contextual information of human activities from sensory information. Another research direction is to record user experiences and reactions to a database for later reference or analysis. An automatic annotation mechanism should be a key issue in such research. In addition, sensing technologies are fundamental parts of both types of systems for capturing human activities and recognizing user attention/intention.

To capture human activities, various modalities are solely or jointly used. These include audio information, visual information, body motions (locations, gestures) and physiological information (e.g., heart rate, perspiration and breathing rate). Consequently, various sensory devices, such as microphones, video

N. Sebe et al. (Eds.): HCI/ECCV 2004, LNCS 3058, pp. 165–175, 2004.

cameras, gyro sensors, GPS (global positioning systems) and skin conductance sensors will be used for detecting this information. Some of these devices are already small and portable enough to wear; however, others still pose difficulties for actual use.

Here, we consider wearable camera systems. In human perception, visual information plays a significant role. Therefore, many systems have employed head-mounted cameras to record images from the user's viewpoint. Using head-mounted cameras, we can easily capture images in a similar view field to the images humans actually perceive. Here, a change of the view field (head motion) reflects a change of the user's attention (except for eye movement). Therefore, this is a very useful device for recognizing user interests.

On the other hand, head-mounted cameras are problematic for users to put on/take off. They cause fatigue to the user's head due to their weight. In addition, their large visual impact makes them difficult to use in daily life. Therefore, it is desirable for a system to have high usability and less visual impact. To achieve this, we propose a body-attached camera system that can record images corresponding to human head motions without requiring the users to wear cameras on their head. By using our system, the user's head motions can be detected in addition to capturing user view images. This is very useful for recognizing not only the orientation of the user's attentions but also the user's head gestures.

In the next section, we briefly summarize related works. Section 3 provides the system overview. Section 4 describes the image processing algorithm. In Section 5, we give experimental results for pose estimation accuracy and show examples of attention region extractions. In Section 6, we address some application domains of the proposed system and delineate future direction. Finally, we conclude this paper in Section 7.

2 Extraction of User's Attention Using Sensory Devices

Here, we briefly summarize related works regarding the recording of human activities using sensory devices.

Personal view images are useful for storing and/or recognizing the user's contextual situation. Therefore, many systems use a wearable camera to capture the user's view image[3, 4]. In these systems, the status of the attention target and surrounding environment are stored as video images in sequence. As mentioned above, visual information constitutes a major part of human experience. This leads researchers to store human experience as video images captured by head-mounted cameras [5, 6].

We proposed a system to estimate human head position and postures from multiple static cameras and head-mounted cameras [7]. In this system, we can extract interaction events among multiple persons from the estimated head motions. This is useful for annotating image data based on user interaction. In this system, however, head pose detection depends on global human positions estimated from multiple static cameras. This means that human attention cannot be extracted without static cameras.

For a different approach from head-mounted cameras, Starner et al. proposed the 'gesture pendant' system [8]. This system utilizes a camera that hangs from the user's neck like a pendant. The camera is used for hand gesture recognition. Healey et al. proposed the 'StartleCam' system [9]. This system can capture the user's frontal view image when the user is startled by using a body-attached camera. In these systems, the user's head motion has not been considered.

Thus, in this paper, we propose a system that does not require users to put cameras on their head by estimating human head motion using an IR (infra-red) camera that looks up. A frontal view camera can be used for extracting the attention region of users from its images in sequence. In the next section, we provide an overview of our system.

3 System Overview

Figure 1 shows the appearance of our system. Our system consists of two parts: a sensing part and a signal processing part.

The sensing part has an IR illuminator, two cameras (one wide-angle CMOS color camera and one IR camera) and a microphone. These parts are usually mounted around the middle of the upper body (as shown in Figure 1). The IR illuminator and IR camera look upwards. The IR illuminator illuminates the user's head (mainly the jowl part) and the IR camera captures the image of the illuminated parts. The wide angle color camera has higher resolution (1280×1024 pixels) than normal video cameras and captures the frontal direction image. A microphone is used for capturing audio data (human voices, environment sounds). According to the property of the CMOS device used in the frontal view camera, we can selectively retrieve a partial image with higher transfer speed. This feature is useful for extracting the user's view image.

The processing part receives signals from the sensing part and estimates head motion using the IR camera image. Then, the image area that is the closest approximately to the human view is selected from the frontal-view camera image based on the head motion estimation results. This flow is described in Figure 2.

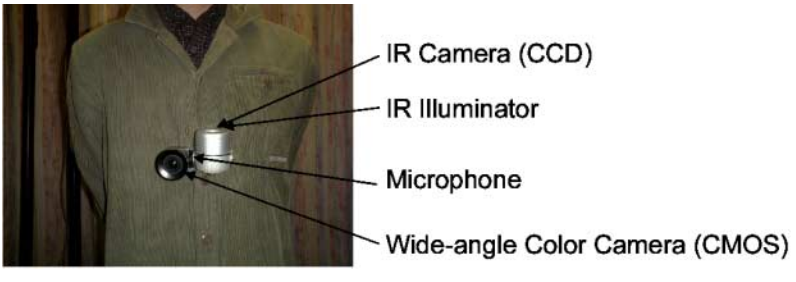


Fig. 1. System Appearance

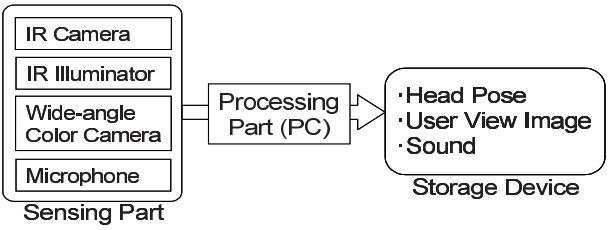


Fig. 2. Process Flow

Using the processing results, we can simultaneously record audio data, image data of the user's view and head motions (gestures).

In the next section, we describe details of the image processing algorithms.

4 Head Pose Estimation & User's View Extraction

4.1 Human Head Model

Before describing our algorithm, we will define the coordinates used in this paper. First we define body-centered coordinates X, Y and Z as shown in Figure 3. Since the direction of the user's view does not depend on rotation about the Z axis, we consider the rotations of two axes, X and Y, only. Here α and β denote the rotations about the X axis and Y axis, respectively. We assume the center of the head rotations is located near the front end of the user's neck.

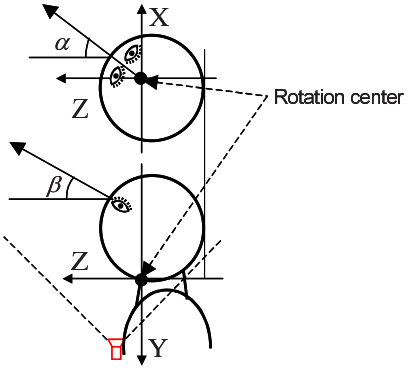


Fig. 3. Head Model

4.2 Human Region Extraction

In our system, we extract the human region (head and torso) by using the IR camera and IR illuminator. The IR camera can shoot only the part illuminated by the IR illuminator. We control the IR illuminator to illuminate only the human region (in a close range) and extract human regions by selecting larger value pixels in the IR camera images (Figure 4). The combination of IR camera and IR illuminator has been used for hand gesture recognition by Numazaki et al.[10].

After the human region extraction, we determine a boundary line between head and torso regions using human region histograms. Here, we determine the center of head rotation to be the mid-point of the boundary line between the head and torso regions. Then, the nose top position in the head region is selected by maximizing the distance between a point and the center of the head rotation (Figure 5). Figure 6 shows some examples of the described process. Here, $\times$ denote estimated positions of nose top points and lines show the estimated head-torso boundaries.

4.3 Head Pose Estimation

In this section, we describe our algorithm for estimating head poses from IR images. Figure 7 describes the relationships among image features. Here, we represent the nose top point and the head rotation center in the IR images as $\boldsymbol{P}_n(=[u_n, v_n])$ and $\boldsymbol{P}_c(=[u_c, v_c])$, respectively. The u axis corresponds to the estimated boundary line between head and torso (Figure 7). v is the perpendicular axis to the u axis.

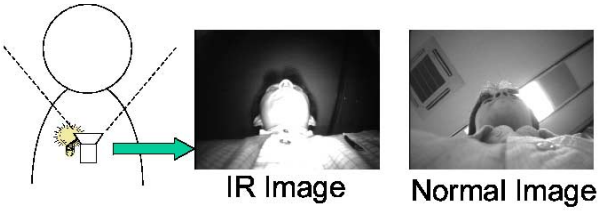


Fig. 4. Human Region Extraction

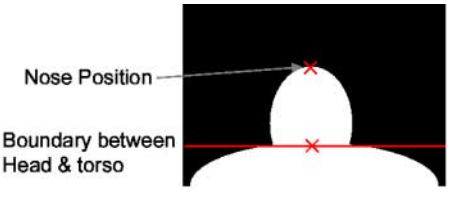


Fig. 5. Feature Detection

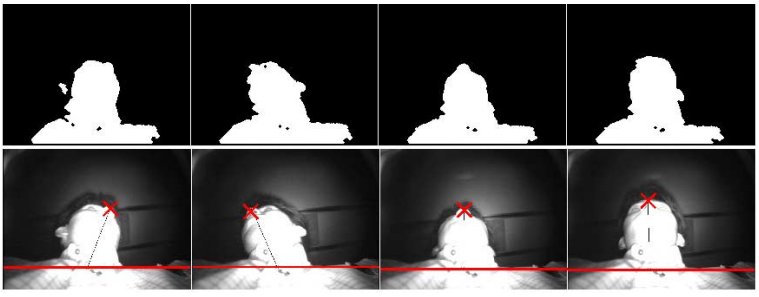


Fig. 6. Human Region Extraction (top: Extracted human regions, bottom: IR images and extraction results)

We assume the head rotation center is located at the front end of the user's neck. Then, head rotation angles α and β can be calculated as follows.

$$\alpha = \tan^{-1}\left(\frac{u_n}{v_n}\right), \tag{1}$$

$$\beta = \cos^{-1}\left(\frac{\sqrt{u_n^2 + v_n^2}}{k}\right) - \cos^{-1}\left(\frac{\sqrt{(u_n^{(0)})^2 + (v_n^{(0)})^2}}{k}\right), \tag{2}$$

where k is a constant value determined by head size and intrinsic parameters of the IR camera, and $\boldsymbol{P}_n^{(0)}$ is the value of $\boldsymbol{P}_n$ when $\alpha = 0$ and $\beta = 0$.

In practice, the relative pose between the camera system and a human body can be changed due to self-rotation (fluctuation) of the camera system. This makes the estimation values of the real pose α and β shifted. We estimate the offset values ($\Delta\alpha$ and $\Delta\beta$) and compensate the self-rotation. $\Delta\alpha$ is determined as the tilt angle of the boundary line (Figure 7). $\Delta\beta$ can be calculated as follows.

$$\Delta\beta = \tan^{-1}\left(\frac{v_c^{(0)} - v_c}{f}\right), \tag{3}$$

where $\boldsymbol{P}_c^{(0)}$ is the value of $\boldsymbol{P}_c$ when $\Delta\alpha = 0$ and $\Delta\beta = 0$. f is the focal length of the IR camera.

4.4 Extraction of User's View

Using the results of the head pose estimation, we can extract an image region corresponding to the user's view from frontal camera images.

Using estimated angles α and β, we can determine the 2D point in a frontal camera image that corresponds to the center of the user's view as follows.

$$x_\alpha = f \cdot \tan(\alpha + \Delta\alpha), \quad y_\beta = f \cdot \tan(\beta + \Delta\beta). \tag{4}$$

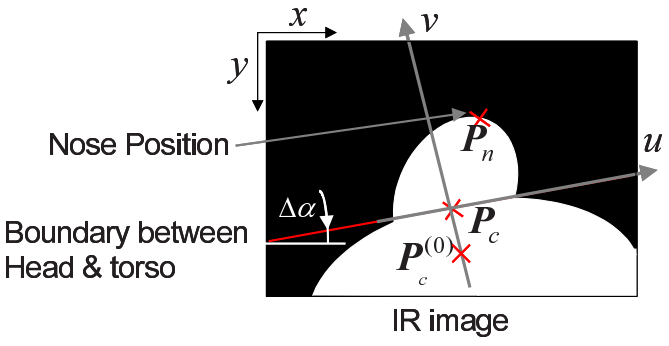


Fig. 7. Head Pose Estimation

Strictly speaking, as the frontal camera is mounted at a different position from human eyes, there is a gap between the extracted view and a real user view. This gap becomes larger when the observed objects are located closer to human eyes. It is hard to solve this problem completely; however, we can reduce the impact of the problem by introducing a non-linear mapping between the head rotation and view selection. This is left for future work.

5 Experiments

To confirm the effectiveness of the proposed system, we performed the following experiments.

First, we evaluated the accuracy of the nose position extraction. We performed nose position extraction with our system and compared the results with the manually selected nose positions. Figure 8 shows the results (Here, solid lines denote the positions estimated with our system and dashed lines denote the ground truth (by manual selection)). As can be seen, nose top points are properly located with our system. The estimation errors here are less than 10 pixels.

Next, we performed head pose estimation using our system. Figure 9 shows the results. Here, solid lines correspond to the trajectory of the estimated gaze points. In the sequence, a subject who wore our system mainly looked at three objects in the scene (a clock, a calendar and a chair).

Based on the duration of a stational head pose, we can extract the image regions that the user is interested in. Figure 10 shows the attention regions automatically extracted from the frontal camera images. In this example, the regions related to the three objects above are properly selected.

As can be seen, our system can successfully detect the changes of user attentions.

Figure 11 shows the user activities as a time sequence obtained in this experiment (from the above captured user's view images, head poses and audio

information). By using this information, we can extract the user's head gestures (nodding, shaking, etc.), and the moment when the user is talking.

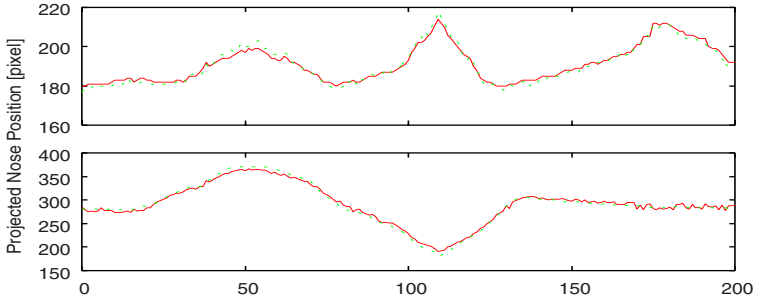


Fig. 8. Nose Position Estimation Accuracy

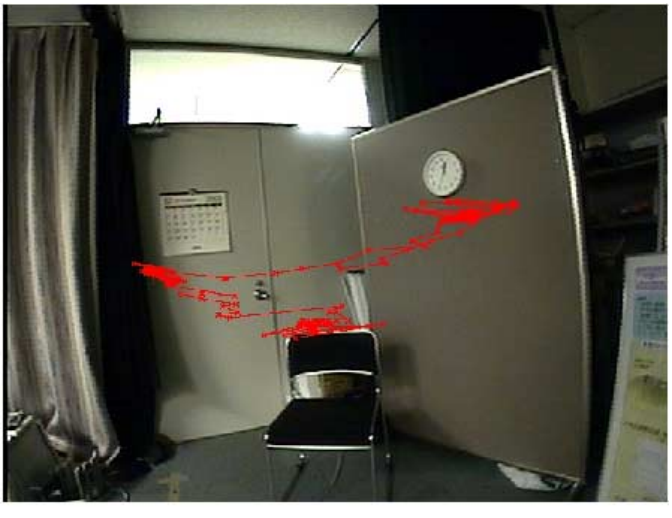


Fig. 9. Estimated Gaze Trajectory

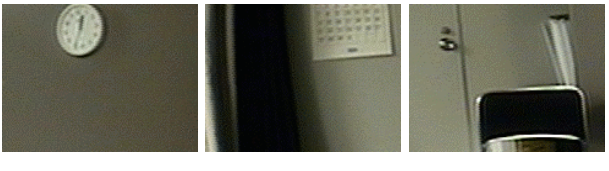


Fig. 10. Extracted User Attention Area

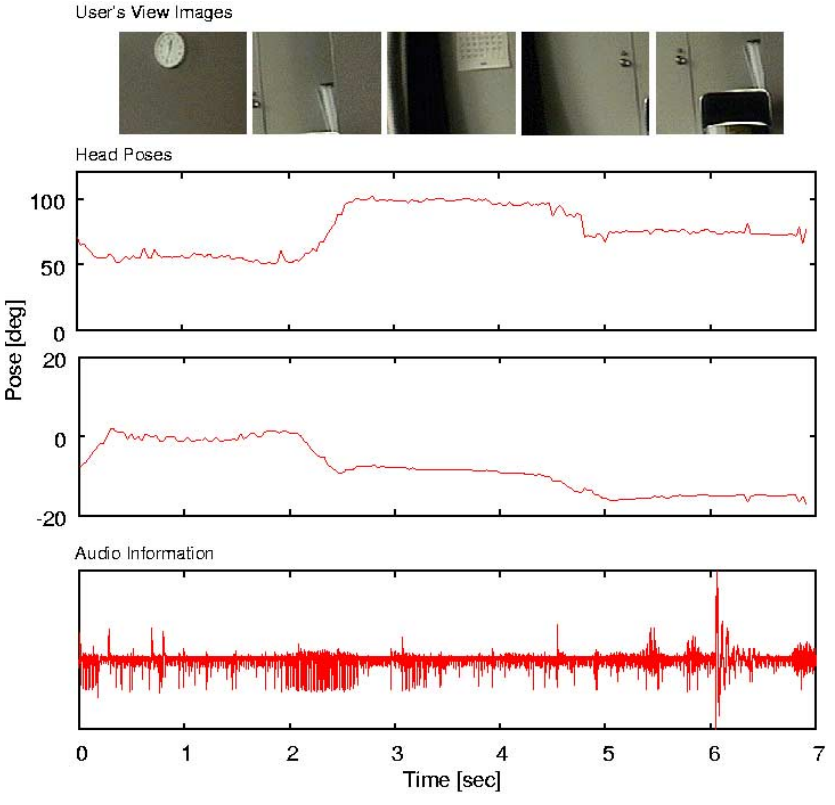


Fig. 11. Captured User's Activities

6 Discussion and Future Direction

In our system, as the user wears the cameras on his/her chest, the user suffers less fatigue and the visual impact to third persons is less than with head-mounted cameras. The system is easy to put on and take off, and is suitable for capturing human daily activities.

Stored data contains useful information for summarizing user experience. Using the head motion data and temporal changes of the image itself, we can extract the user's attention. In experiments (Section 5), no object recognition is considered and just the time length is used to extract the user's attention. Therefore, the system fails to extract attention information if the user is moving. To overcome this problem, we plan to evaluate the similarity of images. Some color histogram-based matching will be considered for this purpose.

Using the proposed system, image sequences approximately in the user's view and head motion information can be stored with audio data. Head gestures can be recognized using head motion information. These data are analyzed for

detecting predefined interactive events, extracting user interests, etc. In addition, head gestures (nodding, shaking, etc.) can be recognized from head motion data. This property of the system is useful for interaction analysis [7].

The prototype system is still insufficient in terms of its size and weight. However, our system has a simple structure and further miniaturization is not difficult. Future versions will become much more compact. Our system lacks global positioning capability though some relative user motions can be estimated from frontal camera images. Position information is helpful for enhancing the quality of video annotation and interaction analysis. We plan to integrate location sensors such as gyro sensors and GPS with our system for that purpose.

7 Conclusion

We proposed a wearable system to capture audio and visual information corresponding to user experience. Using our system, audio information, head motions and images in the user's view are easily recorded in sequence. The system is easy to put on and take off and has less visual impact to third persons. These properties are desirable for capturing human daily activities. We confirmed the effectiveness of our system through experiments.

Future work includes improvement of the head-motion tracking algorithm, head gesture recognition and miniaturization of the entire system. We will also address the analysis of human activities based on user location and content of captured images.

This research was supported in part by the Telecommunications Advancement Organization of Japan.

References

[1] Lamming, M., Flynn, M.: Forget-me-not intimate computing in support of human memory. Technical Report EPC-1994-103, RXRC Cambridge Laboratory (1994) 165

[2] Mann, S.: Wearable computing: A first step toward personal imaging. Computer **30** (1999) 25–32 165

[3] Starner, T., Schiele, B., Pentland, A.: Visual contextual awareness in wearable computers. In: Proc. of Intl. Symp. on Wearable Computers. (1998) 50–57 166

[4] Clarkson, B., Mase, K., Pentland, A.: Recognizing user context via wearable sensors. In: Proc. of Intl. Symp. on Wearable Computers. (2000) 69–75 166

[5] Sumi, Y., Matsuguchi, T., Ito, S., Fels, S., Mase, K.: Collaborative capturing of interactions by multiple sensors. In: Ubicomp 2003. (2003) 193–194 166

[6] Aizawa, K., Shiina, M., Ishijima, K.: Can we handle life-ling video? In: Int. Conf. Media Futures. (2001) 239–242 166

[7] Yamazoe, H., Utsumi, A., Tetsutani, N., Yachida, M.: Vision-based human motion tracking using head-mounted cameras and fixed cameras for interaction analysis. In: Proc. of Asian Conference on Computer Vision 2004. (2004) 682–687 166, 174

[8] Starner, T., Auxier, J., Ashbrook, D., Gandy, M.: Gesture pendant: A self-illuminating, wearable, infrared computer vision system for home automation control and medical monitoring. In: Proc. of Intl. Symp. on Wearable Computers. (2000) 87–94 167

[9] Healey, J., Picard, R. W.: Startlecam: A cybernetic wearable camera. In: Proc. of Intl. Symp. on Wearable Computers. (1998) 42–49 167

[10] Numazaki, S., Morishita, A., Umeki, N., Ishikawa, M., Doi, M.: A kinetic and 3d image input device. In: Proc of CHI'98. (1998) 237–238 169

An AR Human Computer Interface for Object Localization in a Cognitive Vision Framework

Hannes Siegl, Gerald Schweighofer, and Axel Pinz

Institute of Electrical Measurement and Measurement Signal Processing
Graz University of Technology, Austria
{siegl,gschweig,pinz}@emt.tugraz.at
http://www.emt.tugraz.at/~tracking

Abstract. In the European cognitive vision project VAMPIRE (IST-2001-34401), mobile AR-kits are used for interactive teaching of a visual active memory. This is achieved by 3D augmented pointing, which combines inside-out tracking for head pose recovery and 3D stereo HCI in an office environment. An artificial landmark is used to establish a global coordinate system, and a sparse reconstruction of the office provides natural landmarks (corners). This paper describes the basic idea of the 3D cursor. In addition to the mobile system, at least one camera is used to obtain different views of an object which could be employed to improve e.g. view based object recognition. Accuracy of the 3D cursor for pointing in a scene coordinate system is evaluated experimentally.

Keywords: 3D interaction device, active cameras, real-time pose computation, augmented reality, mobile system, mobile AR

1 Augmented Reality

Augmented reality applications enrich perceived reality by giving additional information. This information is provided by representations ranging from text information and object highlighting to the projection of complex 3D objects. Therefore, this technique is perfectly suited as a visual aid for medical and military purposes, for entertainment, for assembly processes or for engineering design or for interactive teaching of visual active memory described in this paper.

Existing AR applications are too limited by restricted mobility and insufficient tracking (head-pose calculation) capabilities to be used in fully mobile, potentially outdoor applications.

The mobile augmented reality system (MARS) by Höllerer *et al.* [3] utilizes an inertial/magnetometer orientation tracker(Intersense) and a centimeter-level / real-time kinematic GPS position tracker which is dependent on a base station providing correction signals. The Tinmith system by Piekarski and Thomas [5] is based on GPS for position tracking and on a magnetometer for orientation tracking.

Our AR-kit has been designed for modular and flexible use in mobile, stationary, in- and outdoor situations. We suggest a wearable system which consists of

N. Sebe et al. (Eds.): HCI/ECCV 2004, LNCS 3058, pp. 176–186, 2004.

two independent subsystems, one for video augmentation and 3D visualization, the other one for real-time tracking fusing vision-based and inertial tracking components.

2 AR Components

An AR-kit usually consists of components providing information on the direction of view – i.e. (self-) localization and head pose recovery – such as vision-based tracking or inertial tracking devices and a possibility for the visualization of information – normally a head mounted display (HMD). Besides, a human computer interface is commonly used for the communication with the system providing the augmented information, for instance the PIP introduced in [8]. In figure 1 the sketch of the system designed for the EU Cognitive Vision project VAMPIRE is shown.

A laptop is used for rendering information and serving the HMD with the video stream captured from a stereo pair consisting of two fire-wire cameras. A custom CMOS camera and an inertial tracker are used for hybrid tracking. A mouse (buttons only) is used as user interface.

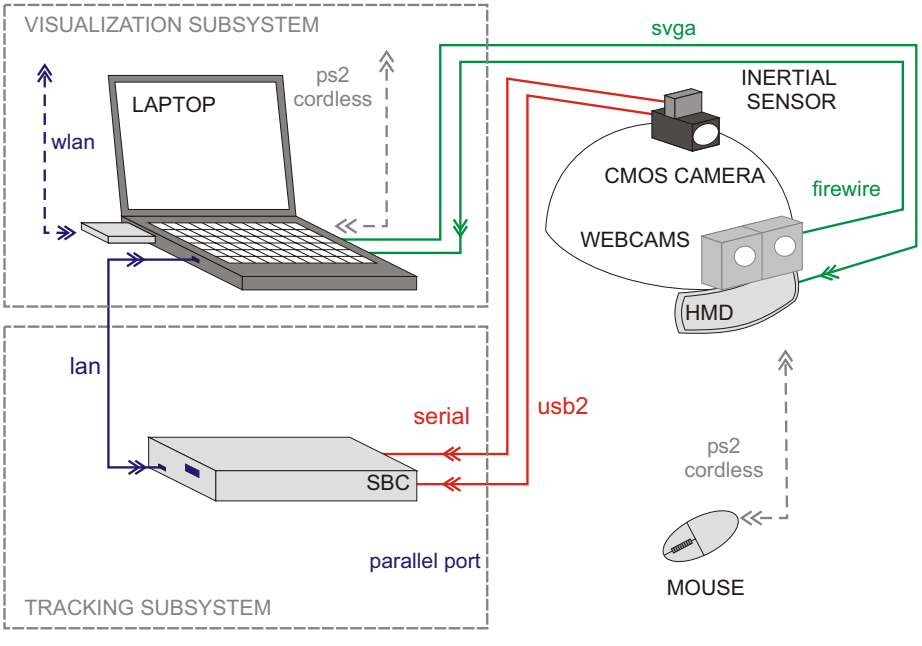
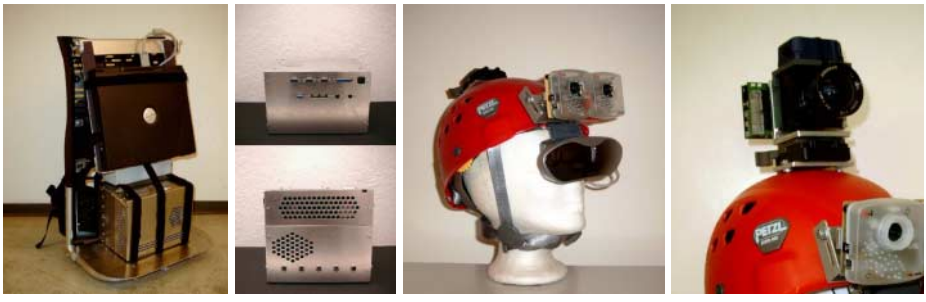


Fig. 1. AR-kit components: A high end laptop and a custom stereo video see-through HMD are employed for visualization. An inertial sensor and a custom high speed CMOS camera are used for tracking



(a) (b) (c) (d)

Fig. 2. AR-kit: Our custom stereo video see-through set consisting of Fire-i firewire webcams and an I-visor 4400VPD HMD(a), hybrid tracking unit consisting of custom CMOS camera and an XSens MT9 (b), the backpack with laptop for visualization (c), snapshots of our custom SBC case (d)

Laptop and single board computer (SBC) are mounted on a backpack (see fig. 2.a and fig. 2.b) and are connected via LAN (direct link). HMD and tracking sensors are mounted on a helmet (see fig. 2.c and 2.d).

2.1 Visualization Subsystem

The laptop applied for visualization has an OpenGL graphic chip (nVidia Quadro) which allows for hardware supplied stereo rendering of the graphics for the custom stereo optical see-through head mounted displays (HMD) consisting of low cost, off-the-shelf components such as two Fire-i firewire webcams and an I-visor 4400VPD HMD (see fig. 2.a). Table 1 lists components and their most important features.

2.2 Tracking Subsystem

A custom mobile PC system has been assembled for hybrid tracking, as laptops seemed to be not flexible enough to allow for experiments with various hardware

Table 1. Components for video loop

Laptop	Dell Precision M50	Pentium 4, 1.8 Mhz, nVidia Quadro4 500 GoGL
HMD	I-visor 4400VPD	SVGA(stereo), 60, 70 and 75 Hz VESA, 31 degree diagonal field of view
Webcams	Fire-i	IEEE1394, 640 × 480, 30 fps (YUV411), 15 fps (RGB, monochrome)

Table 2. Components for hybrid tracking

CPU	Intel PIII	1.2 GHz, Intel socket370
Single Board	Advantech PCI-9577FG	USBII, Gigabyte Ethernet
HD	IBM Microdrive	1 GB
CMOS camera	'i:nex'	1024×1024 pixels, 10 bit, USBII
Inertial sensor	Xsens MT9	6 degrees of freedom
Battery pack	custom	13500 mAh, $\approx$ 1 hour system uptime

Table 3. Request rates vs. window sizes and number of windows: Request denotes a cycle consisting of window positioning and read-out

window side length	number of windows	requests/second
8	5	2600
8	15	2000
8	25	1300
16	5	2000
16	15	1000
16	25	660

for tracking (PCI extensions for e.g. frame grabbers). The system basically consists of a single board computer and a power supply (AC / DC) which also serves all the peripheral hardware of the mobile AR system such as HMD, webcams, CMOS camera and inertial sensor (see tab. 2 for details).

This hardware is mainly used for self-localization or inside-out tracking. We implemented a custom Fuga 1000 based CMOS camera ('i:nex') [4] with USB2 interface to gain extremely fast access of small, arbitrarily positioned image portions (see tab. 3) typically used for tracking of e.g. corners or other local features with small support regions. In order to deal with fast movements of the head, vision-based tracking is fused with a commercially available inertial sensor by Kalman filtering.

3 VAMPIRE System Design

The project "Visual Active Memory Processes and Interactive REtrieval" (VAMPIRE) aims at the development of an active memory and retrieval system in the context of an Augmented Reality scenario. The AR gear provides the image data perceived from the user (stereo camera), the user's head pose (inside-out tracker) and basically the human computer interface (HCI) defining actions (query, learning, naming of objects). The VAM hierarchically maintains the acquired data (image data, head pose, object locations, gesture interpretation) from all the

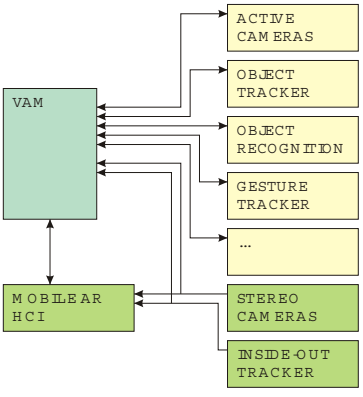


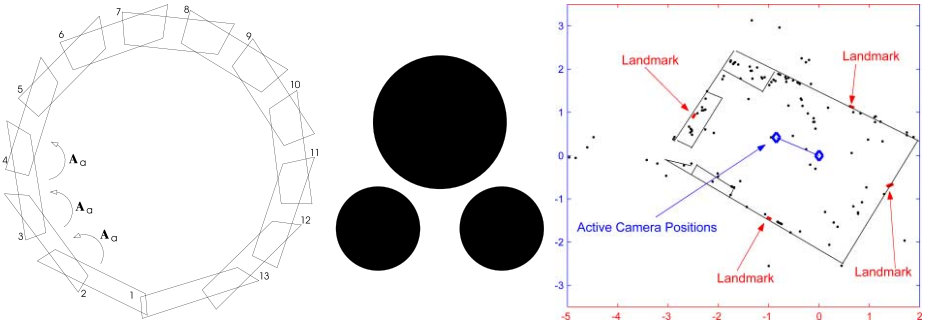
Fig. 3. Vampire System (functional sketch): The visual active memory (VAM) manages and interprets data collected from various modules. The highlighted modules (mobile AR HCI, stereo cameras and inside-out tracker) are for technical reasons physically and functionally closer connected to the VAM than the other modules

connected modules, tries to build contextual relations (cup – coffee – sugar) and thus provides data for connected modules (see fig. 3).

4 VAMPIRE Application Scenario

Within the VAMPIRE project, we aim at mobile indoor applications in unprepared rooms. During an off-line initialization phase the scene is analyzed by recording two image panoramas with a camera (Sony DFW-VL500) mounted on a pan tilt unit (Directed Perception PTU-46-17.5) and extracting a set of artificial landmarks consisting of three disks (see fig. 4.b) and prominent natural corners. This is followed by a sparse reconstruction (see fig. 4.c) of the scene in terms of these landmarks and their scene coordinates using the 'stereo' information provided by multiple recordings [6].

At the moment an artificial target providing corner features (see fig. 5.a) is applied for initialization of the vision-based tracking and the alignment of the coordinate systems (this target defines the origin of the scene coordinate system). Afterwards, the landmarks found during the reconstruction stage are used for online real-time tracking of camera / head pose. Then, the user receives visual feedback using the stereo head-mounted-display (HMD), so that the real scene can be augmented by virtual content. In order to teach the VAM as well as to receive interpretations of the scene and recognition results, several modalities of user-system interaction are required. Pointing at objects in 3D plays an essential role. We found that a 3D cursor [7] would be adequate for most of our applications when an HCI is required for teaching or query, especially in cluttered



(a) (b) (c)

Fig. 4. This figure shows a sketch of the office panoramic view (maximum horizontal rotation of the PTU is 317°) recorded with the camera mounted on the PTU (a). The distance of two different positions of the camera is used as baseline for stereo reconstruction with these panoramas. In order to match these sets of views, a set of targets was applied which do not infer with the natural features (corners) founding the reconstruction of the room. One of these targets is depicted in (b). To the right, an example of a sparse reconstruction of an office scene is shown (c)

scenes. In the subsequent sections the concept of the 3D cursor is explained and an application with an active camera is outlined.

5 Pointing at Objects Using a 3D Cursor

The implemented 3D cursor is basically operated by mouse wheel and buttons. It exploits disparity and object size to generate the perception of distance which allows – together with the head pose obtained from the tracking subsystem – to compute an estimate of an objects size and its position in the room (see fig. 5).

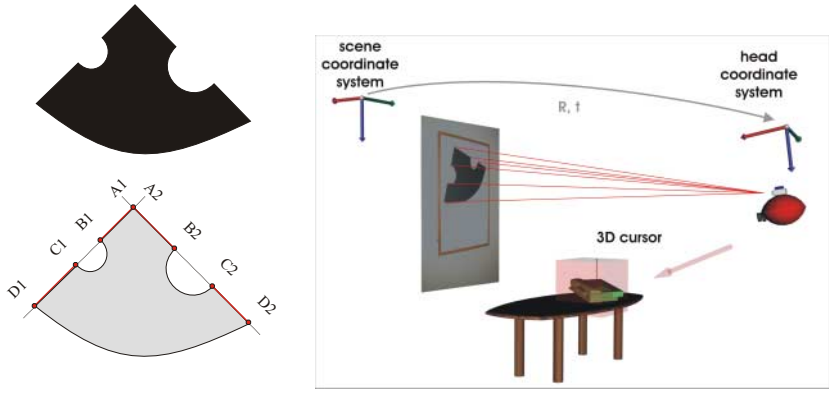
This concept is outlined in figure 6. A horizontal displacement of the cursor from the center of the image planes emulates distance. Hence, point correspondence for stereo vision is established manually by manipulating the x-coordinates of a pair of corresponding points with the mouse wheel.

As the cursor is rather placed in the center of the image than in the border regions where the image is stronger affected by lens distortion, it was tried to calibrate the 3D cursor directly (uncalibrated cameras).

For this purpose, the stereo camera was directed towards objects in 3D with known distance (3 times for each distance) and then the cursor was placed on the object in both images provided by the cameras (see fig. 7).

The mean disparity (see fig. 8) obtained from this localization procedure was used to approximate the relation between distance and disparity by a hyperbola $t_z(d)$

$$t_z(d) = \frac{a}{d+b} \tag{1}$$



(a) (b)

Fig. 5. This figure shows an artificial corner target [1] which is used as an intermediate step on the way to natural features and to initialize vision-based tracking using corners as features, respectively. The target is identified by the perspectively invariant cross ratio (CR) of the segments on the two intersecting lines. The pose can be calculated by the positions of the corners (a). To the right, the 3D cursor application is depicted. The tracking system processes the corner features of the CR target for self-localization. The selection of the phone with a 3D cursor allows to estimate the position of this object in scene coordinates (b)

where a and b are constants determined by a LSE fit ($a = 28.8868$, $b = 1.9464$). In figure 8.b this approximation is compared to the results obtained from the standard stereo vision procedure [2] for the reconstruction of depth (internal and external camera calibration, relative orientation, 2D point correspondence,

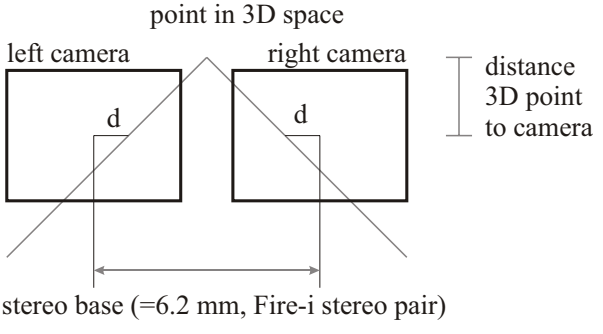


Fig. 6. 3D cursor: A horizontal displacement of the cursor from the center of the image planes emulates distance

Fig. 7. 3D cursor: An object is focused by the user. The object is perceived in the center of the image displayed by the HMD. In fact, there is a displacement in the images provided by the two cameras. For calibration, both images are displayed next to each other and the cursor is moved from the center parallel to the horizontal axis (same disparity for both cameras and images, respectively) until in both images the same position in the scene is covered. This eliminates the deviations caused by users of the stereo HMD

etc). Due to the manual calibration technique (see fig. 7), the difference between the more precise stereo reconstruction method and the direct method increases with the measured distance (decrease of disparities), as the cursor cannot by placed manually that accurately.

Then, the obtained depth or translation along the z-axis of the camera (perpendicular to the image plane) can be written as

$$\mathrm{t}_{disp} = \begin{pmatrix} 0 \\ 0 \\ t_z \end{pmatrix}. \tag{2}$$

Applying the obtained function, it is possible to compute the approximate position of the object in 3D by

$$\mathrm{t}_{obj} = \mathrm{t}_{h2w} - \mathbf{R}_{h2w}\mathrm{t}_{c2h} + \mathbf{R}_{h2w}\mathbf{R}_{c2h}\mathrm{t}_{disp} \tag{3}$$

where t_{obj} approximates the position of the object, $\mathbf{R}_{h2w}$ and t_{h2w} denote the pose received from inside-out tracking and $\mathbf{R}_{c2h}$ and t_{c2h} the relative pose (obtained from extrinsic calibration of all three cameras) of the Fire-i cameras for the video loop, respectively (see fig. 9).

Figure 10 shows an experimental verification of our approach. Three users without any experience with our 3D cursor and one well trained user placed the 3D cursor 2 times on the surface of an object (distance=1,2,..,5 m). It can be seen that the achieved accuracy depends on training and distance to the object.

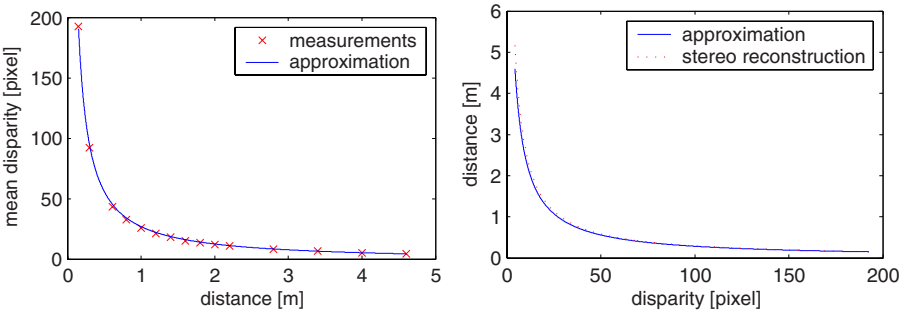


Fig. 8. Calibrating the 3D cursor: Approximation with hyperbola (a) approximation vs. standard stereo vision procedure (b)

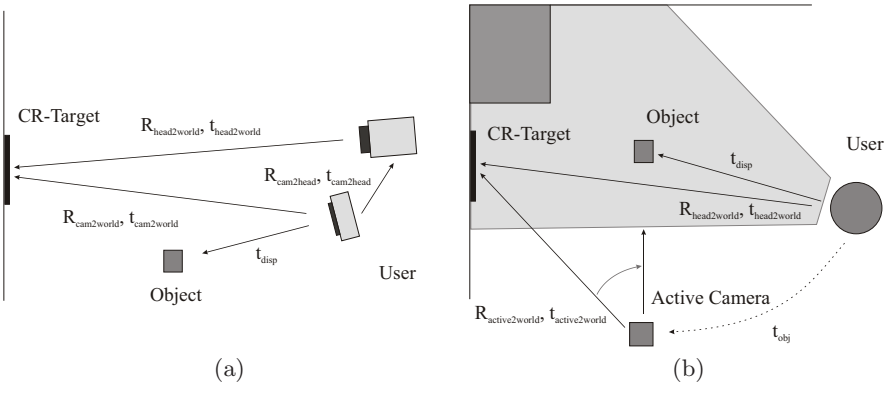


Fig. 9. 3D Cursor in combination with an active camera: Sketch of the required coordinate transforms to compute the external calibration of stereo cameras and tracking camera and to obtain the position of an object in 3D; side view depicting cameras mounted on the mobile Ar kit (a), top view including the active camera and the field of view of the user (b)

6 A 3D Cursor Application

In our lab we implemented the following setup for the 3D cursor (see fig. 11): In the sparsely reconstructed office, the user points at an object in the room using the 3D cursor. His direction of view determined by inside-out tracking and the distance of the object along this direction are used to estimate the absolute position of the object in the room. This information is sent to the computer controlling the active camera which was used to create the sparse reconstruction of the room and uses the same coordinate system as the inside-out tracker because of the CR-target. Therefore, it is possible to change the

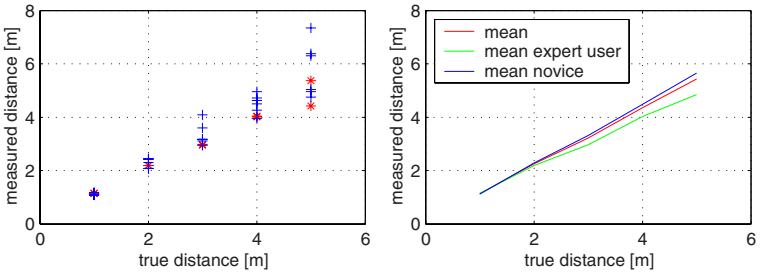


Fig. 10. Experimental verification: Three users (+) without any experience and one well trained user (*) placed the 3D cursor 2 times on the surface of an object (distance=1,2,..,5 m). It can be seen that the achieved accuracy depends on training and distance to the object

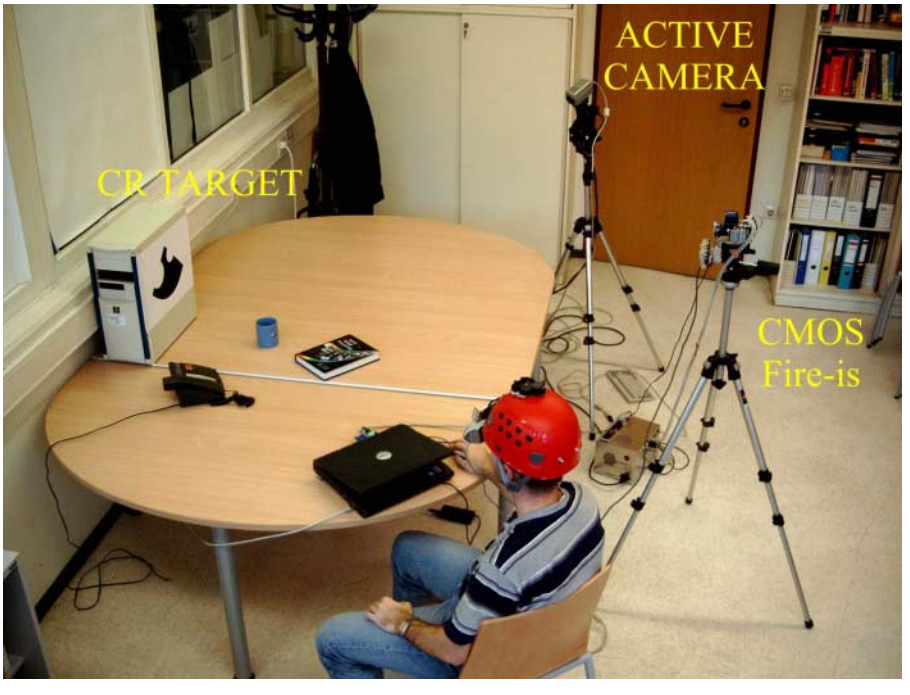


Fig. 11. This figure shows a setup for the verification of our approach. The cameras are mounted on a tripod to eliminate a possible influence of the movements of the user who wears the AR helmet and places the 3D marker next to various objects on the table (error in depth (t_z) < 10 % for a trained user, if $t_z < 4$ m)

direction of view of the active camera so that an independent second view is obtained (in fact, an arbitrary number of cameras could be used). There is a number of applications to this scenario, for instance:

- acquire a different view of a remote object for view based object recognition
- display enlarged images of remote objects in the HMD, e.g. the title of a book on a high bookshelf.

7 Conclusion

We presented the mobile AR gear which is employed as human computer interface for the cognitive vision project VAMPIRE which tries to model human memory processes in an office environment. Besides, we discussed an AR 3D cursor for pointing and presented a 3D cursor application where the object position determined by head pose and the estimated distance via 3D cursor are used in combination with active cameras. In the future, the integration of pointing gestures will yield a more natural feel for simple scenes than the employment of a mouse as interaction device. Besides, various experiments are performed to find the most suitable shape of the cursor.

Acknowledgement

This research was funded by *VAMPIRE* Visual Active Memory Processes and Interactive REtrieval (EU-IST Programme IST-2001-34401), and by the Austrian Science Fund (FWF project S9103-N04).

References

[1] M. K. Chandraker, C. Stock, and A. Pinz, *Real-time camera pose in a room*, 3rd Intern. Conference on Computer Vision Systems, April 2003, pp. 98–110. 182

[2] O. D. Faugeras, *Three-dimensional computer vision : a geometric viewpoint*, MIT Press, 1993. 182

[3] T. Höllerer, S. Feiner, T. Terauchi, G. Rashid, and D. Hallawa, *Exploring MARS: developing indoor and outdoor user interfaces to a mobile augmented reality system*, Computers and Graphics **23** (1999), no. 6, 779–785. 176

[4] U. Mühlmann, M. Ribo, P. Lang, and A. Pinz, *A new high speed CMOS camera for real-time tracking applications*, Proc ICRA 2004, New Orleans. 179

[5] W. Piekarski and B. Thomas, *Augmented reality with wearable computers running linux*, 2^{nd} Australian Linux Conference (Sydney), January 2001, pp. 1–14. 176

[6] M. Ribo, G. Schweighofer, and A. Pinz, *Sparse 3d reconstruction of a room*, submitted to: ICPR'04, Cambridge, 2004. 180

[7] H. Siegl and A. Pinz, *A mobile AR kit as a human computer interface for cognitive vision*, Proc WIAMIS'04, Lissabon, 2004. 180

[8] Z. Szalavari and M. Gervautz, *The personal interaction panel - a two-handed interface for augmented reality*, Computer Graphics Forum **16** (1997), no. 3, 335–346. 177

EM Enhancement of 3D Head Pose Estimated by Perspective Invariance

Jian-Gang Wang[1], Eric Sung[2], and Ronda Venkateswarlu[1]

[1] Institute for Infocomm Research
21 Heng Mui Keng Terrace, Singapore 119613
{jgwang,vronda}@i2r.a-star.edu.sg
[2] Nanyang Technological University
Singapore 639798
eericsung@ntu.edu.sg

Abstract. In this paper, a new approach is proposed for estimating 3D head pose from a monocular image. The approach assumes the more difficult full perspective projection camera model as against most previous approaches that approximate the non-linear perspective projection via linear affine assumption. Perspective-invariance is used to estimate the head pose from a face image. Our approach employs a general prior knowledge of face structure and the corresponding geometrical constraints provided by the location of a certain vanishing point to determine the pose of human faces. To achieve this, eye-lines, formed from the far and near eye corners, and mouth-line of the mouth corners are assumed parallel in 3D space. Then the vanishing point of these parallel lines found by the intersection of the eye-line and mouth-line in the image can be used to infer the 3D orientation and location of the human face. Perspective invariance of cross ratio and harmonic range are used to locate the vanishing point stably. In order to deal with the variance of the facial model parameters, e.g. ratio between the eye-line and the mouth line, an EM framework is applied to update the parameters iteratively. We use the EM strategy to first compute the 3D pose using some initially learned (PCA) parameters, e.g. ratio and length, then update iteratively the parameters for individual persons and their facial expressions till convergence. The EM technique models data uncertainty as Gaussian defined over positions and orientation of facial plane. The resulting weighted parameters estimation problem is solved using the Levenberg-Marquardt method. The robustness analysis of the algorithm with synthetic data and real face images are included.

1 Introduction

Two different transformations may be used for pose estimation from a single view: perspective or affine. Few researchers have addressed the full perspective problem in pose estimation [20,14,13,19]. Beviridge et al [14] use random-start local search with a hybrid pose estimation algorithm employing both full and weak perspective models: a weak-perspective algorithm is used in raking neighbouring points in a search space of model-to-image line segment correspondences, and a full-perspective algorithm is used to update the model′s pose for new correspondence sets. T. Horprasert et al [13]

N. Sebe et al. (Eds.): HCI/ECCV 2004, LNCS 3058, pp. 187-199, 2004.

employ projective invariance of the cross-ratios of the four eye-corners and anthropometric statistics to estimate the head yaw, roll and pitch. Five points, namely the four eye corners and the tip of nose, are used. The four eye corners are assumed to be co-linear in 3D. This, however, is not exactly true in general. Affine projection is assumed in pose estimation approaches [10,9,24]. Horaud et al [12] developed an iterative algorithm for recovering paraperspective pose. Gee et al [10] achieved a real-time face tracker by utilizing simple feature trackers searching for the darkest pixel in the search window. The unique solution has to be searched by projecting both poses back into the image plane and measuring the goodness of the fit. In their earlier work, Gee and Cipolla [9] used five key feature points, nose tip, the outer eye corners and mouth corners, to estimate the facial orientation. The facial model is based on the ratios of four distances between these five relatively stable features, where the ratios were assumed not to change very much for different facial expressions.

The domain knowledge of human face structure can be advantageously used for pose estimation. In this paper, we study a novel approach that calls upon the notion of the vanishing point to derive a new and simple solution for measuring pose of human head from a calibrated monocular view. The vanishing point can be located using perspective invariance of the cross ratio which in our case happens to be the harmonic ratio. The 3D direction of eye-line and mouth-line can then be inferred from the vanishing point. (The vector defined by the direction of the origin to the vanishing point is also the direction of the parallel space lines [16]). Also, an analytic solution of orientation of the facial plane can be obtained if the ratio of the eye-line and mouth-line segments is known. Furthermore, the 3D coordinates of the eye and mouth corners can be located if one of the lengths of the eye-line and mouth-line segments is known.

A variety of iterative optimisation techniques have already been explored in model fitting by difference research groups, e.g. fitting a linear combination of faces to an image [21,7,15]. The common goal of these approaches is to compute the model parameters yielding a rendering of the model that best resembles the target image. The model parameters used in our pose estimation approach, e.g. the ratio of the eye-line and mouth-line segments, are not exactly invariant with respect to face expression and individual. In order to deal with the variance of the facial model parameters, an EM framework is applied to update the parameters iteratively. The parameters are updated by minimizing the residual errors between the predicted features and the actual features on the image with statistical approach. The initial parameters of the EM algorithm are learned (PCA) from a face database.

2 Pose Determination

2.1 3D Model

The 3D model parameters in our proposal include the four ratios of five world lengths, D_f, D_e, D_m, D_s, D_n (Fig. 1 (c)(d)) and length D_e. The two outer eye corners and two mouth corners are assumed to be co-planar in 3D. These planar features are symmetric about the axis defined by the center point of the two eye corners and two mouth corners. The tip of the nose is assumed elevated at some height D_n above this

plane and to lie on the perpendicular plane through the symmetry axis. The perpendicular projection point, P_b, of the nose tip on the face plane divide the symmetry axis into two segments having a ratio $R_m \equiv D_n/D_f$. Throughout this work, a perspective transformation is assumed. Here we need more model ratios than the method in [9] in order to handle the more general perspective projection cases. The model ratios $R_m \equiv D_n/D_f$, $R_n = D_s/D_f$ and $R_e \equiv D_e/D_f$ were used in [9] under weak perspective assumption. Instead of R_e, in this paper, another one, namely, $r \equiv D_e/D_m$ is used, where D_m is the length of the mouth line segment. The parameter vector of our 3D model is defined,

$$\mathbf{a} = (R_m, R_n, r, D_e)^T \tag{1}$$

2.2 Location of the Vanishing Point

In order to locate the vanishing point stably, we use the perspective invariance of the cross ratio that in particular is the harmonic ratio. An ordered set of collinear points is called a *range*, and the line passing through them is called its axis. A range of four points, e.g. $\{W, X, Y, Z_\infty\}$ is called a *harmonic range* if their cross ratio satisfies the harmonic ratio :

$$[W, X, Y, Z_\infty] = (WY/XY)/(WZ/XZ_\infty) = -1$$

Figure 1(a) shows the classic example of a harmonic range, with X as the mid-point of W and Y. Let $\{A, B, C, D\}$ be four image points in general position as in Figure 1(b). Let P be the intersection of AB and DC, Q the intersection of CB and DA. Such a set of six points $\{A, B, C, D, P, Q\}$ is called a *complete quadrilateral*.

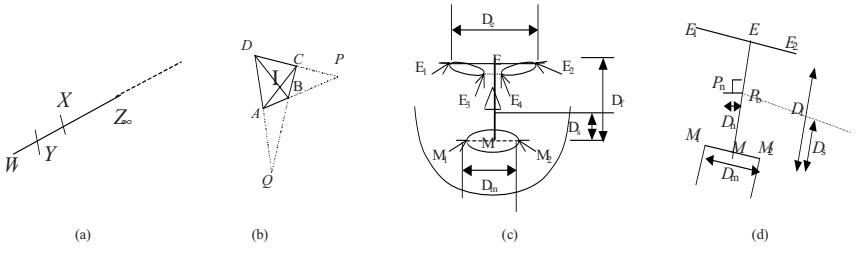


Fig. 1. (a) Cross ratio; (b) complete quadrilateral; (c)(d) 3D model

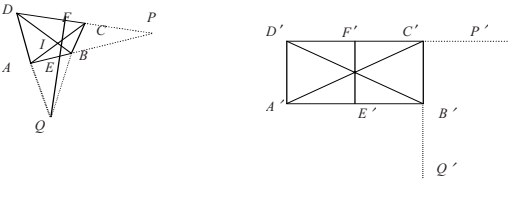


Fig. 2. Mapping four arbitrarily given image points

Proposition Let I be the intersection of AC and BD, E and F be the intersections of IQ with AB and CD, respectively, see Figure 2. Then $\{P, F, D, C\}$ and $\{P, E, A, B\}$ are all harmonic ranges.

Proof of the proposition can be done based on the following propositions and theorem [16].

Proposition 1 A unique collineation is determined that maps four arbitrarily given image points in general position to four arbitrarily given image points in general position.

Proposition 2 Let D' and C' be distinct space point, and let F' be their midpoint, (see Figure 2). If P' is the vanishing point of the space line passing through D' and C', then $\{D', C', F', P'\}$ is a harmonic range.

Theorem 1 The cross ratio is invariant under collineations.

According to proposition 1, four points $\{A, B, C, D\}$ can be mapped to a rectangle by some collineations, refer to Figure 2. D', F', C', P' and Q' correspond to D, F, C, P and Q respectively. P is the vanishing point of P', F' and E' are the midpoints of $D'C'$ and $A'B'$ respectively. $[D', C', F', P']$ is harmonic range according to proposition 2, Hence, $\{D, C, F, P\}$ and $\{A, B, E, P\}$ become harmonic ranges according theorem 1, i.e.

$$[DCFP] = -1 \tag{2}$$

$$[ABEP] = -1 \tag{3}$$

Hence, the vanishing point P can be determined using (2) or (3).

The location of far-eye and mouth corners form the quadrilateral and its equivalent completion is depicted in Figure 1(c). Let E_1 corresponds to D, E_2 corresponds to C, M_1 corresponds to A, and M_2 corresponds to B. We can determine the vanishing point formed by the eye-line E_1E_2 and mouth-line M_1M_2.

2.3 Pose Determination

If the vanishing point is $P(u_\infty, v_\infty)$, (d_x, d_y, d_z) represent the 3D-direction vector of the eye-lines and thus the mouth-line. We have [11]:

$$(d_x, d_y, d_z) = \frac{1}{\sqrt{u_\infty^2 + v_\infty^2 + 1}} \begin{pmatrix} u_\infty \\ v_\infty \\ 1 \end{pmatrix} \tag{4}$$

Let us assume (x_{e1}, y_{e1}) and (x_{e2}, y_{e2}) are normalized image coordinate ($f_x = f_y = 1$) of two far corners of eyes, $E_1(X_{e1}, Y_{e1}, Z_{e1})$ and $E_2(X_{e2}, Y_{e2}, Z_{e2})$ are their 3D coordinate respectively. (x_{m1}, y_{m1}) and (x_{m2}, y_{m2}) are normalized image coordinates of two corners of mouth, $M_1(X_{m1}, Y_{m1}, Z_{m1})$ and $M_2(X_{m2}, Y_{m2}, Z_{m2})$ are their 3D coordinate respectively.

$$X_{ei} = x_{ei}Z_{ei}, \qquad X_{mi} = x_{mi}Z_{mi} \tag{5}$$

$$Y_{ei} = y_{ei}Z_{ei}, \qquad Y_{mi} = y_{mi}Z_{mi} \qquad i = 1, 2.. \tag{6}$$

Generally speaking, the ratio (denoted as r) of the length of the eye-line segment (denoted as D_e) to the length of the mouth-line segment (denoted as D_m) is a more invariant measure than the lengths themselves for different people. We consider two cases. In the first case that r is given instead of the lengths themselves. The ratio is immediately available from a frontal-parallel view of the face. The orientation of the facial plane and relative positions of the corners can be located in the first case. In the second case, both the ratio and one of the lengths are given hence the absolute 3D positions of the four corners and the orientation of the facial plane can be determined.

Case 1: r is known

In this case, the orientation of the facial plane and 3D relative positions of the feature corners can be calculated using the vanishing point and r. From (4), we arrive at

$$((X_{e2}-X_{e1})/D_e, (Y_{e2}-Y_{e1})/D_e, (Z_{e2}-Z_{e1})/D_e)^T = (d_x, d_y, d_z)^T \tag{7}$$

$$((X_{m2}-X_{m1})/D_m, (Y_{m2}-Y_{m1})/D_m, (Z_{m2}-Z_{m1})/D_m)^T = (d_x, d_y, d_z)^T \tag{8}$$

From the definition of r, we have,

$$r = D_e/D_m \tag{9}$$

From (7) to (9), we obtain

$$(Y_{e2}-Y_{e1}, Z_{e2}-Z_{e1})^T = ((d_y/d_x)(X_{e2}-X_{e1}), (d_z/d_x)(X_{e2}-X_{e1}))^T \tag{10}$$

$$(X_{m2}-X_{m1}, Y_{m2}-Y_{m1}, Z_{m2}-Z_{m1})^T = ((X_{e2}-X_{e1})/r, (Y_{e2}-Y_{e1})/r, (Z_{e2}-Z_{e1})/r)^T \tag{11}$$

Let $X_{e2}-X_{e1} = 1$, so $Y_{e2}-Y_{e1}$, $Z_{e2}-Z_{e1}$, $X_{m2}-X_{m1}$, $Y_{m2}-Y_{m1}$, $Z_{m2}-Z_{m1}$ can be determined in turn using (10) to (11). This assumption means we can get only relative 3D positions of the feature corners. However the orientation can be obtained uniquely.

Replace X_{e1} and X_{e2} with (5), Y_{e1} and Y_{e2} with (6), we have

$$X_{e2}-X_{e1} = x_{e2}Z_{e2}-x_{e1}Z_{e1} \tag{12}$$

$$Y_{e2}-Y_{e1} = y_{e2}Z_{e2}-y_{e1}Z_{e1} \tag{13}$$

Z_{e1} and Z_{e2} can be solved from (12) and (13). Hence X_{e1}, X_{e2}, Y_{e1} and Y_{e2} can be solved using (5) and (6). Similarly Z_{m1} and Z_{m2} are found using following equations:

$$x_{m2}Z_{m2}-x_{m1}Z_{m1} = X_{m2}-X_{m1} \tag{14}$$

$$y_{m2}Z_{m2}-y_{m1}Z_{m1} = Y_{m2}-Y_{m1} \tag{15}$$

Hence X_{m1},X_{m2},Y_{m1} and Y_{m2} are found using (5) and (6).

Case 2: D_e or D_m and r are known

In this case, the absolute 3D positions of the feature corners can be determined. Let assume D_e is given. The Eq (7) described in the above case are still be used. Replace X_{e1} and X_{e2} in (7) with (5), we have

$$(x_{e2}Z_{e2}-x_{e1}Z_{e1})/D_e = d_x \tag{16}$$

From (7), we obtain

$$Z_{e1} = Z_{e2} - D_e d_z \tag{17}$$

From (16) and (17), we arrive at

$$Z_{e2} = D_e(d_x - d_z x_{e1})/(x_{e2} - x_{e1}) \tag{18}$$

Replace Y_{e1} and Y_{e2} in (7) with (6), we have

$$(y_{e2} Z_{e2} - y_{e1} Z_{e1})/D_e = d_y \tag{19}$$

From (18) and (19), we obtain

$$Z_{e2} = D_e(d_y - d_z y_{e1})/(y_{e2} - y_{e1}) \tag{20}$$

Z_{e2} and Z_{e1} can be calculated from (18) or (20) and (17). Hence X_{e1}, X_{e2}, Y_{e1} and Y_{e2} are found using (5) and (6).

D_m can be obtained using r and D_e with (9). Similarly, the 3D coordinates of the mouth corners $M_1(X_{m1}, Y_{m1}, Z_{m1})$ and $M_2(X_{m2}, Y_{m2}, Z_{m2})$ can be calculated using their corresponding image coordinates then.

From the relative and absolute 3D coordinates of the four corners obtained under the first and second case respectively, we can calculate the normal **n** to the face plane as the cross product of the two space vectors $\mathbf{M_2E_2}$ and $\mathbf{M_2M_1}$ (see Figure 1(c)):

$$\mathbf{n} = \mathbf{M_2E_2} \times \mathbf{M_2M_1} \tag{21}$$

The four points E_1, E_2, M_1 and M_2 are in general not expected to be coplanar due to noise. So instead, the face normal could be calculated as the average of following cross products of the pairs of space vectors: $\mathbf{M_2E_2}$ and $\mathbf{M_2M_1}$, $\mathbf{E_2E_1}$ and $\mathbf{E_2M_2}$, $\mathbf{E_1M_1}$ and $\mathbf{E_1E_2}$, $\mathbf{M_1M_2}$ and $\mathbf{M_1E_1}$.

A proof of the vanishing line-based pose estimation can be found in [27].

3 Statistical Model Adaptation

3.1 Motivation

It should be noted that the model parameters, e.g. r, with respect to face expressions and individual is not quite invariant especially with significant facial expressions. In this paper, an EM framework is proposed to update the model parameters. Our goal is to adapt the parameters of 3D model that best suit each individual. The 3D head pose with respect the camera is estimated using the vanishing-point based method with ratio and length parameters learned from some 3D face models. Then the 3D coordinates of the middle point of eye-line and the mouth-line (represented with the parameters ratio and length) are perspectively projected to image. The parameters are updated by minimizing the error of the 2D feature location using EM.

3.2 Model Adaptation Using EM Algorithm

The EM algorithm was first introduced by Dempster et al [5] as a means of fitting incomplete data. Our contribution is in the adaptation of the 3D model rather than the transform [26, 3]. Moreover, perspective projection rather than affine transformation [26, 3] is assumed.

3.2.1 Learning the Model Parameters Using PCA

The probabilities of the model parameters can be learned initially from a face database [18, 2]. According to the definition of the model parameter vector, a = (R_m, R_n, r, D_e)^T.

We have available a model of a person′s head as 3D range and texture. Using the 3D model, we can segment out certain area of the face, such as nose, eyes and mouth, etc., as well as determine the location of landmark features (eye and mouth corners, nose tip, etc.) The model ratios and length of a set of training 3D models are then obtained. In this paper, 3D VRML models of XM2VTS DB are used for this purpose. There are 293 3D VRML face models in XM2VTS database. Then we perform PCA on the set of the vectors $\{\mathbf{a}_i\}$ for i=1,..m. We subtract the average $\bar{\mathbf{a}} = \frac{1}{m}\sum_{i=1}^{m}\mathbf{a}_i$ from each model vector, $\bar{\mathbf{b}}_i = \mathbf{a}_i - \bar{\mathbf{a}}$ and define a data matrix $\mathbf{A}$ = ($\mathbf{b}_1$, $\mathbf{b}_2$, .., $\mathbf{b}_m$). The essential step of PCA [6] is to compute the eigenvector $\mathbf{v}_i$, i =1...,m, of the covariance matrix $\mathbf{C} = \frac{1}{m}\mathbf{A}\mathbf{A}^T = \frac{1}{m}\sum_{i=1}^{m}\mathbf{b}\mathbf{b}_i^T$, which can be achieved by a Singular Value Decomposition [22] of $\mathbf{A}$. The eigenvalues of $\mathbf{C}$, $\sigma^2_{a,1} \geq \sigma^2_{a,2...}$ are the variance of the data along each eigenvector. The eigenvectors form an orthogonal basis, $\mathbf{a} = \bar{\mathbf{a}} + \sum_{i=1}^{m-1}\beta_i.\mathbf{v}_i$, and PCA provides an estimate of the probability density within face space [6]: $p(\mathbf{a}) \sim e^{-\frac{1}{2}\sum_i \frac{\beta_i^2}{\sigma_{a,i}^2}}$.

3.2.2 Expectation

Every facial feature can be associated to each of parameters in 3D model with some a *posteriori* probability. The expectation step of the EM algorithm provides an iterative framework for computing the a *posterori* probabilities using Gaussian mixtures defined over the parameters.

The 3D model feature points are represented by coordinate vectors $\mathbf{X_i}$ = (X_i, Y_i, Z_i)^T. We represent each point in the image data set by an augmented position vector u_i = (x_i, y_i, 1)^T. According to the camera model defined in Section 2.2, the perspective projection from 3D to 2D is represented by the following equation

$$u_i(\mathbf{a}) = T(\mathbf{a};\mathbf{X}_i) = \frac{\mathbf{I}_{3\times 3}\mathbf{X}_i}{\mathbf{X}_i^T\Xi} \tag{22}$$

where $\mathbf{I}_{3\times3}$ is an identity matrix, $\Xi = (0, 0, 1)^T$; $\mathbf{a}$ is the model parameter vector that has been defined in (1).

In order to relate the model parameters to the matching, the tip of the nose is estimated based on the pose output and then re-projected onto the image. The error between the projection and the real measurement of the feature is used in EM. According to (22), the 2D coordinates of the nose tip are: $(x_n, y_n)^T = (X_n/Z_n, Y_n/Z_n)^T$.

Let the unit normal to the facial plane be (n_x, n_y, n_z). The coordinate of P_b is (X_b, Y_b, Z_b) as shown in Fig. 1(d). Then the 3D coordinates (X_n, Y_n, Z_n) of the nose tip N can be computed from the eye and mouth corners and the normal of the facial plane.

$$(X_n\ Y_n\ Z_n)^T = (X_b\ Y_b\ Z_b)^T + R_m D_f (n_x, n_y, n_z)^T \tag{23}$$

where $$(X_b\ Y_b\ Z_b)^T = (X_m + R_n(X_e - X_m),\ Y_m + R_n(Y_e - Y_m),\ Z_m + R_n(Z_e - Z_m))^T \tag{24}$$

where $E(X_e, Y_e, Z_e)$ and $M(X_m, Y_m, Z_m)$ are the middle points of the eye and mouth line segments respectively, $D_f = \|EM\|$.

The predicted feature position is $u_i(\mathbf{a}^{(n)})$. Assume the detected feature locations in the image is $\{w_i\}$, the error is then

$$E_i(\mathbf{a}^{(n)}) = (u_i(\mathbf{a}^{(n)}) - w_i) \tag{25}$$

The EM algorithm considers the conditional likelihood for the 2D facial feature locations w_i given the current parameters, $\mathbf{a}^{(n)}$. The algorithm builds on the assumption that the individual data items are conditionally independent of one another given the parameters,

$$p(w|\mathbf{a}^{(n)}) = \Pi_i\, p(w_i|\mathbf{a}^{(n)}) \tag{26}$$

We want to minimize the function,

$$E(\mathbf{a}^{(n)}) = \Sigma_i \|u_i(\mathbf{a}^{(n)}) - w_i\|^2 \tag{27}$$

To minimize the function with respect to **a,** we employ an EM algorithm. Given the feature points $w = \{w_i\}$, the task is to find the model parameters $\mathbf{a}$ with maximum posterior probability $p(\mathbf{a}|w)$. According to Bayes rule,

$$p(\mathbf{a}|w) \sim p(w|\mathbf{a})P(\mathbf{a}) \tag{28}$$

$P(\mathbf{a})$ was estimated with PCA [23]. We assume that the required model can be specified in term of a multivariate Gaussian distribution. The random variables appearing in these distributions are the error residuals for the 2D position predications of the ith template point delivered by the current estimated model parameters. Accordingly, we have

$$p(w_i \mid \mathbf{a}^{(n)}) = \frac{1}{(2\pi)^{4/2}\sqrt{|\Sigma|}} \exp\left[-\frac{1}{2} E_i(\mathbf{a}^{(n)}) \Sigma^{-1} E_i(\mathbf{a}^{(n)})\right] \tag{29}$$

where Σ is the covariance matrix for the vector of error-residuals E_i between the predicated template points and the facial feature location in the image. The

expectation step of the EM algorithm simply reduces to computing of the weighted squared error criterion

$$Q'(\mathbf{a}^{(n+1)}|\mathbf{a}^{(n)}) = -(1/2)\ \Sigma P(\mathbf{a}^{(n)})\ E_i(\mathbf{a}^{(n)})^T \Sigma^{-1} E_i(\mathbf{a}^{(n)}) \quad (30)$$

3.2.3 Maximization

The maximization step aims to update the parameters $a^{(n+1)}$,

$$a^{(n+1)} = \text{argmax}_a\ Q'(a|a^{(n)}) \quad (31)$$

We solve the implied weighted least-squares minimization problem using the Levenberg-Marquardqt algorithm [22]. $P(a)$ was updated using an incremental method [25].

4 Experimental Results

We have tried our algorithm on synthetic data and real image. The experiments show that our algorithm can provide a good estimation of pose of human head within a close distance.

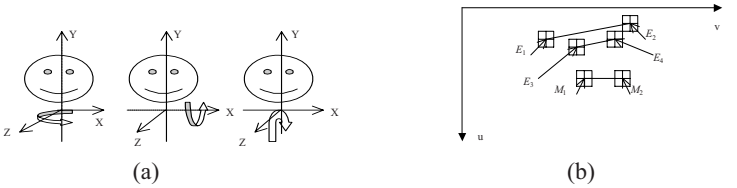
4.1 Simulation Using Synthetic Data

The synthetic data of different poses are produced as in Figure 4(a). We define a 3D rotation by three consecutive rotations around the coordinate axes, that is, a rotation by α degrees around the x-axis first, then a rotation by β degrees around the y-axis, and finally a rotation by γ degrees around the z-axis. The initial coordinates of the four eye and mouth corner points in the target face are set as:

$$E_1(-5.25, -6, Z),\ E_2(5.25, -6, Z),\ M_1(-2.65, -1, Z),\ M_2(2.65, -1, Z)$$

where Z is the distance along the z-axis from the origin to the face. The initial model ratios are set as: $a^{(0)} = (R_m, R_n, r, D_e)^T = (0.6, 0.4, 1.98, 10.5)^T$

The initial 3D coordinate of the tip of the nose and its perpendicular projection point on the face plane can be computed from (35) and (36).



(a) (b)

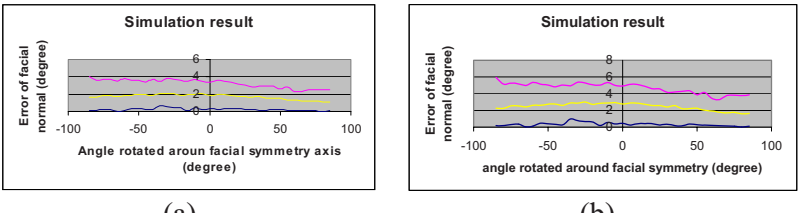
Fig. 3. Simulations. (a) Simulations for different poses; (b) Adding perturbations to the facial corners on the image plane perturbations *n* pixels to a corner point means the new position of the corner will lie at random within the corner-centred (2n+1)×(2n+1) window positions

We show an example of the simulation pose results generated by rotating the facial plane about the facial symmetry (see Figure 3(a) for details), and the perturbation applied to the four eye and mouth corners of 1 pixel standard deviation each as in Figure 3(b), the perturbation applied to the three model ratios (R_m, R_n, r) of 0.1 standard deviation each, to the model length of standard deviation 1. Four corner points rotated about the face symmetry axis from left -80^0 to 80^0 in steps of 5^0. 100 simulation results are generated and averaged. The errors of the facial normal estimation are shown in Figure 4. We can see also that the closer the camera gets to the human face, the more accurate the estimations and this can be seen by comparing Figure 4(a) and Figure 4(b). The error is found to be less than 2^0 when the distance between the original facial plane and the image plane is 60cm. The performance is better than the performance reported in [3] (error was found to be 3^0 for moderate rotation angles) where orthographic projection is assumed. The errors of the 3D positions of the four corners are shown in Figure 5. The error is found to be less than 0.2 cm for moderate rotation angles when the distance between the original facial plane and the image plane is 60cm. The degenerate case (when the facial plane is roughly parallel to image plane) occurs in the angle range of $(-2^0, 2^0)$. The degenerate case is detected when the eye-lines and mouth line are nearly parallel in the image.

4.2 Experiment with Real Images

We have evaluated our algorithm on video sequence of face images, as a person is moving his head in various directions in front of a PC. The experiments show that our method has a good performance and is robust for pose estimation of human head.

Figure 6 shows some experimental results obtained with a plastic model of a head. The convergences of the tip of the nose and its perpendicular projection point on the face plane are shown in Figure 6. The top row of Figure 6 show the initial position of the feature points and rotation angles about the vertical axis: ground truth; the bottom of Figure show the final positions of the feature points after the convergence of the EM algorithm and the estimated rotation angles about the vertical axis. Some frames from a sequence of a subject are shown in Figure 7, where the facial normals of a face image are represented as arrows shown under the face image.



(a) (b)

Fig. 4. Errors of the facial normal when the perturbation applied to the four corners is 1 pixel. Three curves, top: maximum errors; middle: mean errors; bottom: minimum errors. The distance between the original facial plane and the image plane is: (a) 50 cm; (b) 60 cm

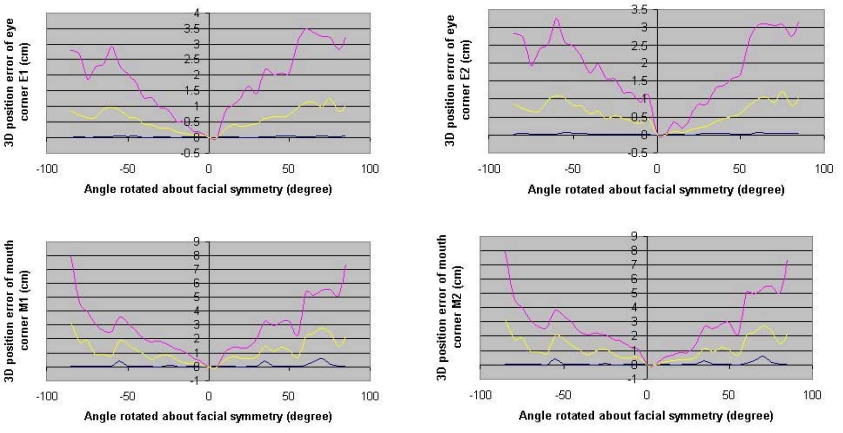


Fig. 5. 3D position errors of the four corners, top left: E_1; top right: E_2; bottom left M_1; bottom right: M_2. The distance between the original facial plane and the image plane is 60 cm

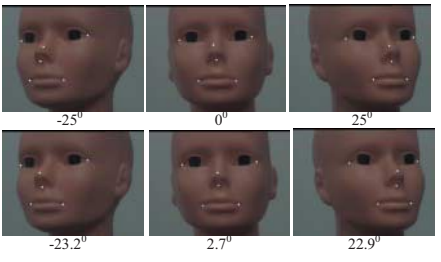


Fig. 6. The convergence of the tip of the nose and its perpendicular projection point on the face plane, top row: initial step, bottom row: final step; The rotation angles about the vertical axis, top row: ground truth, bottom row: estimated

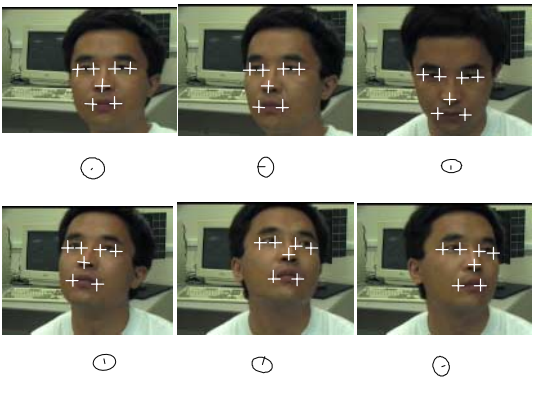


Fig. 7. Pose determination results of some frames from a sequence

5 Conclusion

In this paper, we have presented a new method for computing head pose by using projective invariance of the vanishing point. An analytic solution has been derived and the pose can be determined uniquely when the ratio of the length of the eye-line segment to the length of the mouth-line segment is known. Our algorithm is reliable because the 3D model parameters are iteratively updated using EM approach and thereby allowing for the algorithm to adapt to any individual. The approach assumes the more difficult full perspective projection camera model as against most previous approaches that use the affine assumption. The robustness analysis shows that it is an viable alternative approach for estimating 3D pose (position and orientation) from a single view, especially when an automatic method of finding the vanishing point is possible. Furthermore, our algorithm is reliable because the ratio we used here is more invariant over the human face ensemble than the use of the lengths themselves.

Accuracy of the vanishing point computation plays an important role on performance of the proposed method. The vanishing point can often be obtained from the image itself by some standard techniques [1, 17, 8], and so making our algorithm practical. In situations where the distance between the face and camera is close, the full perspective model that we used can provide more accurate pose estimation than other existing methods that are based on the affine assumption.

References

[1] Almansa, A.Desolneux, S.Vamech : Vanishing point detection without a prior information, IEEE Transactions on PAMI, 25(4) (2003) 502-507.

[2] V. Blanz, S. Romdhani and T. Vetter, Face identification across different pose and illumination with a 3D morphable models, In Proceedings of IEEE International Conference on Automatic Face and Gesture Recognition, (2002) 202-207.

[3] K. N. Choi, M. Carcassoni and E.R. Hancock, Estimating 3D facial pose using the EM algorithm, In H. Weschsler, P. J. Phillips, V. Bruce, F. F. Soulie and T.S. Huang, Editors, Face Recognition From Theory to Application, Springer, (1998) 412-423.

[4] A. D. J. Cross and E. R. Hancock, Graph matching with a dual-step EM algorithm, IEEE Transactions on PAMI, 20(11) (1998) 1236-1253.

[5] A. P. Dempster, N.M. Laird and D.B. Rubin, Maximum-likelihood from incomplete data via the EM algorithm, Journal Royal Statiscal Soc. Series B (methodological), 39, (1977) 1-38.

[6] R. O. Duda, P. E. Hart and D.G. Stork, Pattern Classification, second ed. John Wiley & Sons, 2001.

[7] G. J. Edwards, C.J.Taylor and T.F. Cootes, Interpreting face image using active appearance models, In Proceedings of IEEE Conference on Automatic Face and Gesture Recognition, (1998) 300-305.

[8] A.C. Gallagher. A ground truth based vanishing point detection algorithm. Pattern Recognition, 35 (2002) 1527-1543.

[9] A. Gee and R. Cipolla. Estimating gaze from a single view of a face. In Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, (1994) 758-760.

[10] A. Gee and R. Cipolla. Fast visual tracking by temporal consensus. Image and Vision Computing, 14 (1996) 105-114.

[11] R. M. Haralick, L. Shapiro. Computer and Robot Vision. Addison-wesley publishing company, 1993
[12] R. Horaud, F. Dornaika, B. Loamiroy, and S. Christy, Object pose: the link between weak perspective, para–perspective and full perspective, International Journal of Computer Vision, 22 (1997) 173-189.
[13] T. Horprasert, Y. Yacoob and L. S. Davis. Computing 3-D Head orientation from a monocular image sequence. In Proceedings of IEEE Conference on Automatic Face and Gesture Recognition, (1996) 242-247.
[14] J. R. Beviridge and E.M. Riseman, Optimal geometric model matching under full 3D perspective, Computer Vision and Image Understanding, 61 (1995) 351-364.
[15] M. J. Jone and T. Poggio, Hierachical morhable models, In Proceedings IEEE International Conference on Computer Vision, (1998) 820-826.
[16] K. Kanatani, Geometric computation for machine vision, Clarendon Press, Oxford, (1993).
[17] 17. M. J. Magee and J. K. Aggarwal. Determining vanishing points from perspective images. Computer Vision, Graphics and Image Processing, 26 (1984) 256-267.
[18] B. Moghaddam and A. Pentland, Probilistic visual learning for object representation, IEEE Transactions on PAMI, 19(7) (1997) 696-710.
[19] M.J. Black and Y. Yacoob, Recognizing facial expressions in image sequence using local parameterised models of image motions, International Journal of Computer Vision, 25(1) (1997) 23-48.
[20] P. David, D. Dementhon, R. Duraiswami and H. Samet, SoftPOSIT : simultaneous pose and correspondence determination, In Proceedings of ECCV, (2002) 698-714.
[21] F. Pighin, R. Szeliski and D. H. Salesin, Resynthesizing facial animation through 3D model based tracking, In IEEE International Conference on Computer Vision, (1999) 143-150.
[22] W. H. Press, B. P. Flannery, S.A. Teukolsky and W.T. Vetterling, Numerical Recipes in C: The art of Scientific Computing, 2nd Edition: Cambridge University Press: Cambridge, UK, (1992).
[23] T. S. Jebara and A. Pentland, Parametrized structure from motion for 3D adaptive feedback tracking of faces, In IEEE International Conference on CVPR, (1997) 144-150.
[24] B. Tordoff, W. W. Mayol, T. E.de Campos and D.W. Murry, Head pose estimation for wearable robot control. In Proceedings of British Machine Vision Conference, (2002) 807-816.
[25] P. Hall, D. Marshall and R.Martin, Merging and splitting eigenspace models, IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(9) (2000) 1042-1049.
[26] W. M. Wells, Statistical approaches to feature-based object recognition, International Journal of Computer Vision, 21(1/2) (1997) 63-98.
[27] J.-G. Wang, Head pose and eye gaze estimation for human-machine interaction, PhD thesis, Nanyang Technological University, Singapore, (2001).

Multi-View Face Image Synthesis Using Factorization Model

Yangzhou Du and Xueyin Lin

Department of Computer Science and Technology
Tsinghua University, Beijing 100084, P. R. China
dyz99@mails.tsinghua.edu.cn
lxy-dcs@tsinghua.edu.cn

Abstract. We present a sample-based method for synthesizing face images in a wide range of view. Here the "human identity" and "head pose" are regarded as two influence factors of face appearance and a factorization model is used to learn their interaction with a face database. Our method extends original bilinear factorization model to nonlinear case so that global optimum solution can be found in solving "translation" task. Thus, some view of a new person's face image is able to be "translated" into other views. Experimental results show that the synthesized faces are quite similar to the ground-truth. The proposed method can be applied to a broad area of human computer interaction, such as face recognition across view or face synthesis in virtual reality.

1 Introduction

Given a photograph of a person's face, we can imagine its appearance in other viewpoints. This task, however, is rather difficult to perform with computer. The appearance of face in 2D image can change dramatically as the view angle varies, for the visible region may move to a new position and the visibility of face region also changes. In addition, this variability is slightly different from person to person, since human faces share a common structure but with little differences. There have been considerable discussions of synthesizing face image in novel views which can be roughly divided into two categories: those based on 3D head model and those based on 2D image statistics.

The most natural way to synthesize novel views of a face is to recover its 3D structure. However, accurate 3D reconstruction from 2D images has proved quite difficult or even behaved as an ill-posed problem. In practice many techniques for "3D face modeling" impose prior knowledge on facial measurement or rely on manual assistance of matching a deformable face model to the images. When only one face image is available, the texture within occluded region becomes undefined. Vetter et al. [1, 2] use the linear object class approach to deal with the problem. It is assumed that a new face's texture can be represented as a linear combination of the texture from a group of example faces in the same view, and the combination coefficients can be used to synthesize the face image in another view. Similar to this idea, later they built a morphable face model [3] by exploiting the statistics of a large dataset of 3D face scans. Given one or more face images, the model can be optimized

N. Sebe et al. (Eds.): HCI/ECCV 2004, LNCS 3058, pp. 200-210, 2004.

along with a set of parameters such that it produces an image as close as possible to the input image.

2D methods can also produce promising results for novel view face synthesis. The technique of View Morphing [4] is an extension to image morphing that correctly handles 3D projective camera and produces a virtual viewpoint of face. Because the technique relies exclusively on image information, it works well only when the most face region are visible in both source images. While building statistical models for face images, Cootes et al. [5] found that certain model parameters are responsible for head pose changes. Later they combine a number of 2D linear models to capture the shape and appearance of a face from a wide range of viewpoints. It is known as "view-based active appearance models" [6] and can be used to predict unseen views of a face from view previous seen. In the rest of the paper, we call the "unseen" view as the new view, and previous seen view the old one for convenience. Okada et al. [7, 8] propose a parametric piecewise linear subspace method and demonstrate its application for analyzing and synthesizing facial images with a wide range of pose variation. Though only multi-view wavelet feature images are presented in their report, it is straightforward to make an extension to whole face synthesis.

Face image changes their appearance due to different sources of variation. These influence factors are entangled to form an observed image. The factorization model provides a way to figure out each factor which enables correct understanding and manipulating face image. Tenenbaum and Freeman [9] showed its capacity by wonderful results of face recognition across pose and face synthesis under different illumination. Bregler et al. [10] use such a model to handle the interaction between facial expression and visual speech. Recently factorization model has attracted more attention and has been applied to data analysis and synthesis. It has been found its extension to probabilistic framework [11] and its relevant methodologies such as three-mode PCA [12] and N-mode SVD [13]. We have also made our contribution to nonlinear-extension of the factorization model in our previous work [14].

In this work we present a novel method of multi-view face synthesis that is based on factorization model. It appears as a pure 2D statistical method without any reference of 3D head structure. Here a face image is regarded as an observation in high dimensional space that is influenced by “human identity” and “head pose”. A factorization model is trained with a face database so that it can handle the interaction between the two influence factors. Original bilinear factorization model is generalized into nonlinear case so that it can derive the optimum result while solving the “translation” task. Our experimental results demonstrate how some view of a new person's face can be translated into other views, providing only a single face image. Our method can be applied to areas such as face recognition across view or face synthesis in virtual reality.

This paper is organized as follows. In Section 2, we will review the original bilinear factorization model and its capability of solving the “translation” task. Section 3 briefly introduces the principle of kernel-based nonlinear extension and then gives the nonlinear form of factorization model. Section 4 shows the face database used in the model training and specifies our parametric representation of the face image. The experimental results are presented in Section 5, together with the

problem encountered and our current solution. Finally we conclude our work in Section 6.

2 The Bilinear Factorization Model

In the bilinear factorization model proposed by Tenenbaum and Freeman [9], the two influence factors are generically referred to as "style" and "content". For an observation data set, the contribution of style s is denoted as a I-d vector $\mathbf{a}^s$ and that of content class c as a J-d vector $\mathbf{b}^c$. If $\mathbf{y}^{sc}$ is used to denote a K-d observation vector in style s and content c, then its k-th component can be expressed as a bilinear function of $\mathbf{a}^s$ and $\mathbf{b}^c$,

$$y_k^{sc} = \sum_{i=1}^{I}\sum_{j=1}^{J} w_{ijk} a_i^s b_j^c = \mathbf{a}^{s^T} \mathbf{W}_k \mathbf{b}^c . \tag{1}$$

Here i, j and k denote the components of corresponding vectors, and w_{ijk} characterizes the interaction of two factors. $\mathbf{W}^k$ is a $I \times J$ matrix with components $\{w_{ijk}\}$, that describe a bilinear map from those style and content vector space to the observation space. The equation can be written in a different vector form,

$$\mathbf{y}^{sc} = \sum_{i=1}^{I}\sum_{j=1}^{J} \mathbf{w}_{ij} a_i^s b_j^c . \tag{2}$$

Here $\mathbf{w}_{ij}$ denotes a K-d vector with components $\{w_{ijk}\}$. From Eq.(2), it can be seen that observation $\mathbf{y}^{sc}$ is produced by a linear combination of a group $I \times J$ basis vectors $\mathbf{w}_{ij}$, with coefficients given by product of components of $\mathbf{a}^s$ and $\mathbf{b}^c$. The original training procedure of factorization model can be conducted with an iterative use of singular value decomposition (SVD).

The factorization model has been used in many tasks, such as classification, extrapolation and translation. In [9] the terminology of "translation" is defined as translating from new content observed only in new styles into old styles or content classes. For translating from a single test vector $\tilde{\mathbf{y}}$, we adapt the trained model simultaneously to both the new content class $\tilde{c}$ and the new style $\tilde{s}$, while holding the interaction weight terms learned during training. If vectors $\mathbf{w}_{ij}$ are stacked into a matrix

$$\mathbf{W} = \begin{bmatrix} \mathbf{w}^{11} & \cdots & \mathbf{w}^{1C} \\ \vdots & \ddots & \\ \mathbf{w}^{S1} & & \mathbf{w}^{SC} \end{bmatrix},$$

Eq.(2) can be rewritten as

$$\mathbf{y}^{sc} = [\mathbf{W}\mathbf{b}^{c}]^{VT}\mathbf{a}^{s}, \tag{3}$$

where $\mathbf{W}^{VT}$ is the vector transpose of $\mathbf{W}$. Referring to Eq. (3), the translation is an optimization problem aiming to minimizing the cost function

$$f(\mathbf{a}^{\tilde{s}}, \mathbf{b}^{\tilde{c}}) = \left\| \tilde{\mathbf{y}} - [\mathbf{W}\mathbf{b}^{\tilde{c}}]^{VT}\mathbf{a}^{\tilde{s}} \right\|^2 \tag{4}$$

respect to $\mathbf{a}^{\tilde{s}}$ and $\mathbf{b}^{\tilde{c}}$. Specifically, we can use the following two equations to estimate parameter vectors $\mathbf{a}^{\tilde{s}}$ and $\mathbf{b}^{\tilde{c}}$ alternatively until converge,

$$\mathbf{a}^{\tilde{s}} = \mathrm{pinv}\left([\mathbf{W}\mathbf{b}^{\tilde{c}}]^{VT}\right)\tilde{\mathbf{y}}, \tag{5}$$

$$\mathbf{b}^{\tilde{c}} = \mathrm{pinv}\left([\mathbf{W}^{VT}\mathbf{a}^{\tilde{s}}]^{VT}\right)\tilde{\mathbf{y}}, \tag{6}$$

where pinv() denotes pseudo-inverse function. Normally, the initial values of $\mathbf{b}^{\tilde{c}}$ are taken from the mean of style vectors in the training set. After the iteration procedure is done, we can predict data of known content c in new style with $y_k^{\tilde{s}c} = \mathbf{a}^{\tilde{s}T}\mathbf{W}_k\mathbf{b}^{c}$, and those of known style s in new content class with $y_k^{s\tilde{c}} = \mathbf{a}^{s^T}\mathbf{W}_k\mathbf{b}^{\tilde{c}}$.

In the experiments of multi-view face synthesis the human identity and head pose are regarded as style and content of face image, respectively. When a new person's face in a new view, corresponds to test vector $\tilde{\mathbf{y}}$ in Eq.(4), is presented here, the factorization model is able to translate old person's faces into a new view, and to translate the new person's face to old view. We found that the iterative procedure of Eq.(5) and Eq.(6) behaves sensitive to initial value in our case. This makes the optimization procedure usually traps into local minimum and fails to give a good solution, as shown in Section 5. Fortunately, this difficulty can be overcome by casting original bilinear model into nonlinear scenario as shown in next section.

3 Kernel-Based Nonlinear Extension

The original factorization model proposed by Tenenbaum and Freeman is in a bilinear form and can be extended into a nonlinear form with kernel methods. Recently, a number of kernel-based learning machines have been proposed, such as SVM [15], kernel Fisher discrimination [16] and kernel PCA [17], their special features and successful applications have been reported for various fields. Kernel-based algorithms make use of the following idea: via a nonlinear mapping

$$\Phi : \mathbf{R}^N \rightarrow \mathbf{F}$$

$$\mathbf{x} \mapsto \Phi(\mathbf{x})$$

input data in $\mathbf{R}^N$ is mapped into a potentially much higher dimensional feature space $\mathbf{F}$. All computations which can be formulated in term of dot products in $\mathbf{F}$ are performed by a kernel function without explicitly working in $\mathbf{F}$, $k(\mathbf{x},\mathbf{y}) = \Phi(\mathbf{x})\bullet\Phi(\mathbf{y})$. In our experiments Gaussian function

$k(\mathbf{x},\mathbf{y}) = \exp\left(-\|\mathbf{x}-\mathbf{y}\|^2 / c\right)$ is selected as the kernel function, where c is equals to the twice sum of the data's variance along each dimensions. One consideration of using this form of function is that it has the advantage of fast finding pre-image of vectors in feature spaces [18] and this feature makes image synthesis related tasks easy to do.

Fig. 1. A person's face varying profile views, together defined feature points across views

Taking the strategy of kernel-based algorithms usually use, we map the observed data in input space nonlinearly to high dimensional feature space and then perform linear factorization in the feature space. The observation vector $\mathbf{y}^{sc}$ is mapped into a feature space $\mathbf{F}$ via a nonlinear mapping Φ first, and then the image of $\mathbf{y}^{sc}$ is still described as bilinear function of style and content parameters in $\mathbf{F}$,

$$\Phi(\mathbf{y}^{sc}) = \sum_{i=1}^{I}\sum_{j=1}^{J}\mathbf{w}_{ij}a_i^s b_j^c \,. \tag{7}$$

Though this equation takes the same form as Eq.(2), it is now working in feature space $\mathbf{F}$ and the corresponding parameters get different values. Specifically, $\mathbf{w}_{ij}$ becomes basic vector in feature space for this case. All of the formulas in original model are now working in feature space except that $\mathbf{y}$ is substituted by $\Phi(\mathbf{y})$. For details of this kernel-based nonlinear extension of the factorization model, the reader is referred to our previous work [14].

4 Face Dataset and Its Representation

The face data used in this experiment were taken from the PIE database [19] for which contains a great deal of face images under a wide range of viewpoints. Fig. 1 shows a person's faces varying from left to right profile views. As there will be results of face synthesis in different views, a proper parametric representation for face image is necessary. We take the same representation as that was used in Active Appearance Model (AAM) [5], where a face image is decomposed into shape and shape-free texture parts. The shape is expressed by a coordinate vector of facial feature points and the texture is described by a series of pixel values after the face is aligned to a reference shape. Principal Component Analysis method (PCA) is adopted to reduce the dimensionalities of the shape and texture vectors, as usually do.

As for pose variation, the self-occlusion problem of the feature points should be concerned. That is, some feature points become hidden behind other facial parts while head's pose changes. Since the dimension of the vectors should be a constant while PCA and the factorization model is performed. Occluded feature points introduce uncertainties both in shape vectors and texture vectors, resulting in missing values for

certain vector components. In [7] this problem is handled by filling each missing component by a mean computed from all available data at the component dimension. This is not a good strategy due to that it introduces a bias not related to the true nature of the data. The method does not usually perform well when the number of missing components becomes large.

Based on this consideration, our definition of facial feature points adopt the similar strategy used in [20] where a multi-view nonlinear active shape model was successfully built. As there is almost no occlusion region in the frontal view, a set of feature points are selected around facial organs. While the head rotates, we keep these occluded points stay on the contour of the face where they are just becoming invisible, as shown in Fig.3. By means of this strategy, it can be seen that the 8 points around left eye overlap to each other in the most right view. Though this location does not correspond to the true locations of these feature points defined in the frontal pose, it still works for manipulating shape and texture variations effectively in 2D image plane. In this way, any person's face in any view is expressed with a vector of constant dimension, and such representation of face is still reversible. In other words, a face image is able to be reconstructed from such kind of parametric vector, for example, the translation result of the factorization model.

5 Experimental Results

In our experiment a dataset of 41 person's face in 9 poses is selected, one of which is shown in Fig. 1. The feature points on faces were detected with Active Shape Models (ASM) search. First the experiments of translation between human identity and head pose by using leave-one-out methodology are discussed. Specially, while one face image is regarded as the testing person with a testing pose, the rest 40 persons' faces with the remaining 8 poses are served as training data. Such a case is shown in Fig. 2. Two more cases for translating head poses to another test person's face are shown in Fig. 3. In order to compare the performance of our method with that of original bilinear method we have done the experiments in both methods. For simplicity, here we call the original factorization model as linear model and our kernel-based model as nonlinear model. The experimental results by using both linear and nonlinear models are shown in these Fig.s. It can be seen that the synthetic faces with nonlinear model are quite similar to the ground-truth, while some of the linearly synthetic ones are terribly poor. We have tested both models for hundreds of times in our experiment, and the superior of the nonlinear one over the linear one has been indicated consistently.

Probing the failure reason of the linear model, we have found that the iterative equations of Eq.(5) and Eq.(6) behave very sensitive to the initial value and the translation results heavily depended on the initial guess for style or content vector. This usually makes the solving procedure trap into a local minimum, and hence fail to give global optimum solution. We have tried the iterative procedure many times with different initial values selected randomly. However, it has been proved by our experiments that a “good” initialization is quite difficult to find. In fact, synthetic faces with linear model plotted both in Fig.2 and Fig.3 are the best results we ever obtained.

We have also tried to solve the problem related to the linear model by means of Adaptive Simulated Annealing (ASA) algorithm [21]. Despite the bilinear nature, we have treated the translation task as general optimization problem, i.e. Eq.(4), and employed ASA algorithm to solve it. Unfortunately, the cost function drops rather slowly as the temperature parameter value goes down, and the ASA procedure seems hard to stop in a couple of hours, since the searching space was rather large. When the procedure was broken enforcedly, the "current best" solution was usually better than that obtained by the iterative equations (5) and (6) with the same initial value. The synthetic results, however, were still not appeared to be accepted faces.

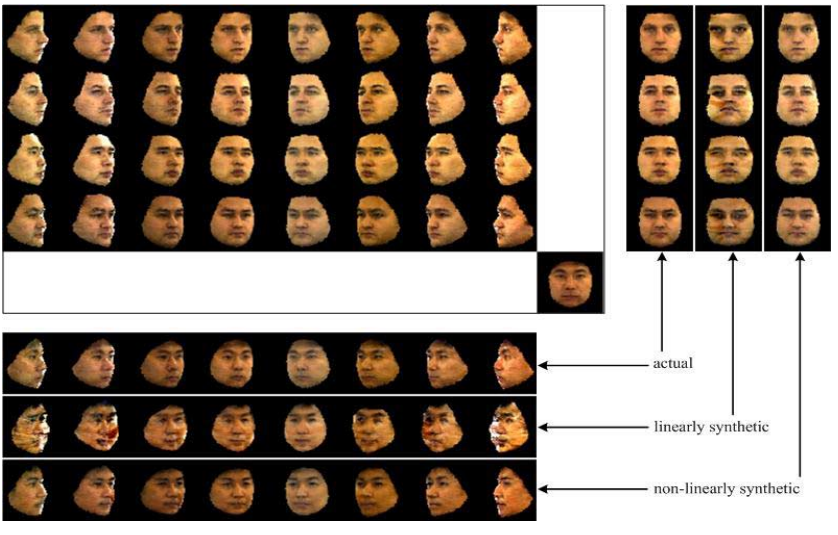


Fig. 2. Translation of face images across human identities and face poses. The four person's faces (style) viewed in eight poses (content) are parts of data used for model training. The test image of a new person's face viewed in a new pose is solely showed next to the bottom-left corner of training images. Actual images, synthesized ones by two kinds of models are labeled with strings "actual", "linearly synthetic" and "non-linearly synthetic", respectively

Such problem encountered in the linear model, however, was never encountered in the nonlinear model using Gaussian kernel, and the originally "stupid" iterative procedure always gives a global optimum solution with arbitrary initialization! It is clearly shown in the Fig. 2. that the "content" (frontal pose) embedded in the testing face has been successfully translated to each person of the training set.

The testing person's faces with different poses translated from the training set look well. In order to examine the synthesis error quantitatively, we calculate the image difference between real faces and nonlinearly synthetic faces by averaging altogether the differences of all three channels. The result is shown in Fig. 4. For display purpose, the error values are added by half of the maximum grayscale value. Here the mean of absolute error within effective face region is 16.6 grayscales. It can be seen that the luminance deviation is mainly concentrated on cheek and forehead while the

Fig. 3. Two more cases for translating a new person's face to the old poses. For each case, the ground-truth, linearly and non-linearly translated faces are shown in the first, second and third rows, respectively. That standing alongside is the test face of a new person with a new pose

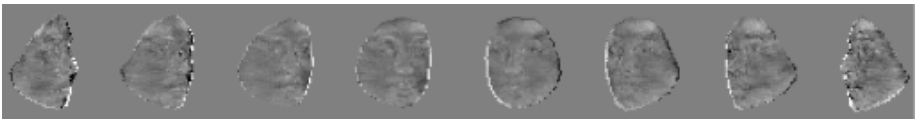


Fig. 4. Differences between actual face and non-linearly synthesized face in Fig.2

Table 1. The place numbers for measuring how the non-linearly translated faces close to the ground-truth. The three rows are listed for the cases in Fig.2, upper and lower parts in Fig.3, respectively. Each of the columns corresponds to a difference pose plotted in the figures

Fig.2	5	2	2	2	1	6	4	3
Fig.3, (1)	3	3	1	11	2	1	1	6
Fig.3, (2)	4	1	2	1	2	5	2	8

inaccuracy on the area of key facial feature, such as eyes, nose and mouth, is relatively small. Other trivial error occurred on the face contour which may arise by inconsistent background.

In order to evaluate the synthesis result in other aspect, we should also examine whether the synthesized face is similar to someone within training set. To do this, the differences of the synthetic face with each face with the same view in the training set are calculated and sort in ascending order. The places synthetic faces hold in the cue are shown in Table 1. In effect, this assess procedure is a face recognition one across a wide range of view. If the number in the table is one, then the person's identification has been correctly recognized with a new pose condition. Though not all of the place numbers are kept very small, the result looks quite cheerful, considering that we are dealing with the difficult face recognition problem of crossing a wide range of pose variation.

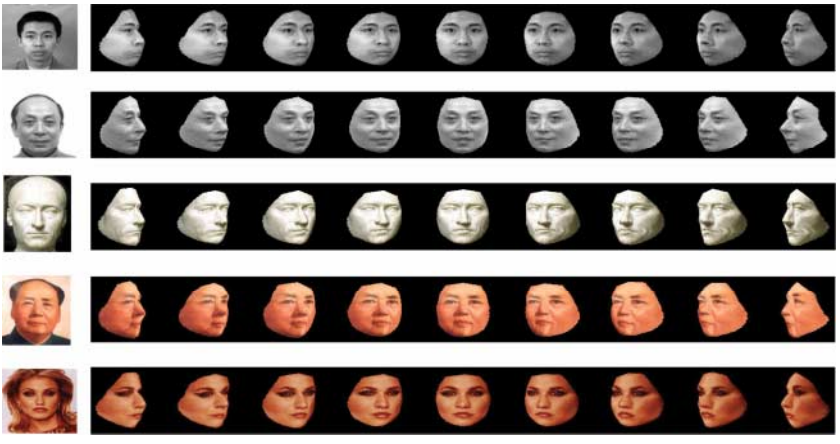


Fig. 5. Multi-view face synthesis for arbitrary person. Left: input image; right: synthesis result

We have also tested the generalization performance of factorization model, especially for nonlinear model. We have tried to synthesize a novel view for an arbitrary person with the trained factorization model directly, but failed to get a satisfying result. Actually our model just deals with two influence factors, i.e., human identity and head pose, it ignores another important influence factor - the illumination. The face images within training set were captured under the same lighting condition. The illumination of an arbitrary face image, however, appears different from that within the training set. No doubt that our factorization model would fail to handle it well. In order to get over this difficulty in the present work, we adopt a relighting technique [22] that can re-render the input face image towards the same illumination effect as that in training faces. This method has been proved to be effective in our experiment.

Another reason causing fail, is due to the limited size of training set. Despite linear or nonlinear model, the synthesized face is eventually a certain combination of training examples, in our case the PIE database. Since all the training faces consist of just 41 person's face in 9 poses, the face space covered by the model is rather limited. Therefore if a new face appearance is far from that in the training set, that new face will be expressed poorly. Our experiments indicated that the quality of the reconstructed facial texture part is poor, while the shape part of an arbitrary face, however, is expressed quite well due to its rather smaller dimensionality. Thus we take only the resultant shape part calculated from the factorization model and warp the texture of the input face image to the desired novel view. Actually this operation tells how far each pixel should move to a new location while translating the input face to a specific viewpoint. By using this strategy, in combination with face relighting technique mentioned above, we got quite good results. Several examples are shown in Fig.5, which looks quite promising. Since image warping cannot handle occlusion region in face, the input face image has to be limited in frontal view.

6 Discussion and Conclusion

In this work a sample-based method for multi-view face synthesis is proposed. A factorization model is used to explicitly handle the interaction between "human identity" and "head pose" that are regarded as the variation source of face appearance in multi-view images. The experimental results show that this method is able to produce a new person's face image in a wide range of view, providing only a single face image. One limitation of current method is that the input image of arbitrary person's face is expected to be near frontal. This would be improved if a better parametric representation of face image is found. The proposed technique can be used to HCI areas such as view independent face recognition or face animation in virtual environment.

Acknowledgements

This research is supported partially by the National Natural Science Foundation of China (No.69975009) and the National Grand Fundamental Research 973 Program of China (No.2002CB312101).

References

[1] Vetter, T.: Synthesis of novel views from a single face image. International Journal of Computer Vision, v 28, n 2, (1998) 103-116

[2] Vetter, T., Poggio, T.: Linear object classes and image synthesis from a single example image. IEEE Transactions on Pattern Analysis and Machine Intelligence, v 19, n 7 (1997) 733-742

[3] Blanz, V., Vetter, T.: Morphable model for the synthesis of 3D faces. In Proceedings of the ACM SIGGRAPH Conference on Computer Graphics (1999) 187-194

[4] Seitz, S.M., Dyer, C.R.: View morphing. In Proceedings of the ACM SIGGRAPH Conference on Computer Graphics (1996) 21-42

[5] Lanitis, A., Taylor, C. J., Cootes, T. F.: Automatic interpretation and coding of face images using flexible models. IEEE Transactions on Pattern Analysis and Machine Intelligence, v 19, n 7 (1997) 743–756

[6] Cootes, T.F., Wheeler, G.V., Walker, K.N., Taylor, C.J.: View-based active appearance models. Image and Vision Computing, v 20, n 9-10 (2002) 657-664

[7] Okada, K., Von der Malsburg, C.: Analysis and synthesis of human faces with pose variations by a parametric piecewise linear subspace method. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, v 1 (2001) I761-I768

[8] Okada, K., Akamatsu, S., Von der Malsburg, C.: Analysis and synthesis of pose variations of human faces by a linear PCMAP model. In Proceedings of the International Conference on Automatic Face and Gesture Recognition (2000) 142-149

[9] Tenenbaum, J.B., Freeman, W.T.: Separating style and content with bilinear models. Neural Computation, vol. 12, (2000) 1247-1283

[10] Chuang, E., Deshpande, H. Bregler, C.: Facial Expression Space Learning, In Proceedings of Pacific Graphics, (2002)

[11] Grimes, D.B., Shon, A.P., Rao, R.P.N.: Probabilistic bilinear models for appearance-based vision. In Proceedings of the IEEE International Conference on Computer Vision, v 2, (2003) 1478-1485
[12] Davis, J.W., Gao, H: Recognizing human action efforts: An adaptive three-mode PCA framework. In: Proceedings of the IEEE International Conference on Computer Vision, v 2 (2003) 1463-1469
[13] Wang, H., Ahuja, N.: Facial expression decomposition. In: Proceedings of the IEEE International Conference on Computer Vision, v 2 (2003) 958-965
[14] Du, Y., Lin, X.: Nonlinear factorization models using kernel approaches. In: Proceedings of the Asian Conference on Computer Vision, v1 (2004) 426-431
[15] Vapnik, V.N.: The nature of statistical learning theory. Springer Verlag, New York (1995)
[16] Baudat, G., Anouar, F.: Generalized discriminant analysis using a kernel approach. Neural Computation, vol. 12, no. 10. (2000) 2385-2404
[17] Schölkopf, B., Smola, A.J., Müller, K.-R.: Nonlinear component analysis as a kernel eigenvalue problem. Neural Computation, vol. 10, (1998) 1299-1319
[18] Mika, S., Schölkopf, B., Smola, A.J., Müller, K.-R., Scholz, M., Rätsch, G.: Kernel PCA and de-noising in feature spaces. In Advances in Neural Information Processing Systems 11, (1999) 536-542
[19] Sim, T., Baker, S. Bsat, M.: The CMU Pose, Illumination, and Expression (PIE) Database. In Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition, (2002)
[20] Romdhani, S., Gong, S., Psarrou, A.: A Multi-View Nonlinear Active Shape Model Using Kernel PCA. In Proceedings of the British Machine Vision Conference, (1999) 483-492
[21] http://www.ingber.com/ASA-README.html
[22] Shashua, A., Riklin-Raviv, T.: Quotient image: Class-based re-rendering and recognition with varying illuminations. IEEE Transactions on Pattern Analysis and Machine Intelligence, v 23, n 2 (2001) 129-139

Pose Invariant Face Recognition Using Linear Pose Transformation in Feature Space

Hyung-Soo Lee and Daijin Kim

Pohang University of Science and Technology
Department of Computer Science and Engineering
San 31, Hyoja-Dong, Nam-Gu, Pohang, 790-784, Korea
{sooz,dkim}@postech.ac.kr

Abstract. Recognizing human face is one of the most important part in biometrics. However, drastic change of facial pose makes it a difficult problem. In this paper, we propose linear pose transformation method in feature space. At first, we extracted features from input face image at each pose. Then, we used extracted features to transform an input pose image into its corresponding frontal pose image. The experimental results show that recognition rate with pose transformation is much better than the result without pose transformation.

1 Introduction

In modern life, the need for personal security and access control becomes important issue. Biometrics is a technology which is expected to replace traditional authentication methods which is easy to be stolen, forgotten and duplicated. Fingerprints, face, iris, and voiceprints are commonly used biometric features. Among these features, face provides more direct, friendly and convenient identification way and is more acceptable compared with individual identification ways of other biometrics features[1]. Thus, face recognition takes one of the most important parts in biometrics. Many researchers have investigated face recognition. However, face appearance in nature scenes varies drastically with changes of facial pose, illumination conditions and so forth. Such variations make face recognition process difficult. Among these difficulties, pose variation is the most critical and challenging one.

Beymer[2], Biuk[3],and Huang[4] divided face images into several subsets according to facial angles and model each view subspace respectively. Then they estimated the pose angle of input facial image and projected the image onto the corresponding subspace. Finally they classified the face image in projected subspace. Such view-based scheme is preferred because it is avoided to explicitly establish 3D model from each pose image, which often tends to be a more complicate problem.

In this paper, we propose linear pose transformation method in feature space. First we compute subspace of each pose images using PCA or kernel PCA. Then we compute pose transformation matrix between input pose subspace and frontal

N. Sebe et al. (Eds.): HCI/ECCV 2004, LNCS 3058, pp. 211–220, 2004.

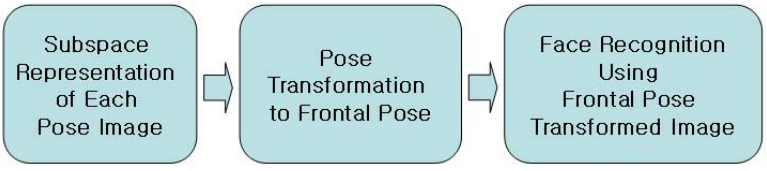


Fig. 1. Process of Pose Transformation

pose subspace. We can represent any input face by a linear combination of basis vectors. If we have a pair of posed facial image and its corresponding frontal facial image for the same person, we can obtain an estimation of the transformation between posed image and frontal image by using the coefficients of training data. Using obtained pose transformation matrix, we can transform any input posed image to frontal posed image. Finally we have images transformed to frontal pose, then we can use usual face recognition method such as LDA[5], GDA[6], NDA[7][8] and nearest neighbor. Fig.1 shows the process of our proposed method.

2 Subspace Representation

Input facial image is generally very high-dimensional data and pose transformation in input space shows usually low performance. Therefore we need to represent facial image in subspace not only for dimensionality reduction, but also for relevant feature extraction. In this section, we review two methods of subspace representation PCA and kernel PCA.

2.1 Principal Component Analysis

From the viewpoint of both the curse of dimensionality and the optimality of the pattern classification, it is desirable to reduce the dimensionality of feature space of the data. In PCA[9], a set of observed n-dimensional data vector $\boldsymbol{X} = \{\boldsymbol{x}_p\}$, $p \in \{1, \cdots, N\}$ is reduced to a set of m-dimensional feature vector $\boldsymbol{S} = \{\boldsymbol{s}_p\}$, $p \in \{1, \cdots, N\}$ by a transformation matrix T as

$$\boldsymbol{s}_p = T^t(\boldsymbol{x}_p - \mathcal{E}[\boldsymbol{x}]), \tag{1}$$

where $m \leq n$, $T = (\boldsymbol{w}_1, \cdots, \boldsymbol{w}_m)$ and the vector $\boldsymbol{w}_j$ is the eigenvector which corresponds to the jth largest eigenvalue of the sample covariance matrix $C = \frac{1}{N}\sum_{p=1}^{N}(\boldsymbol{x}_p - \mathcal{E}[\boldsymbol{x}])(\boldsymbol{x}_p - \mathcal{E}[\boldsymbol{x}])^T$, such that $C\boldsymbol{w}_k = \lambda_k \boldsymbol{w}_k$. The m principal axes T are orthonormal axes onto which the retained variance under projection is maximal. One property of PCA is that projection onto the principal subspace minimizes the squared reconstruction error $\sum \parallel \boldsymbol{x}_p - \hat{\boldsymbol{x}} \parallel^2$. The optimal linear reconstruction of $\hat{\boldsymbol{x}}$ is given by $\hat{\boldsymbol{x}} = T\boldsymbol{s}_p + \mathcal{E}[\boldsymbol{x}]$, where $\boldsymbol{s}_p = T^t(\boldsymbol{x}_t - \mathcal{E}[\boldsymbol{x}])$, and the orthogonal columns of T span the space of the principal m eigenvectors of C.

2.2 Kernel Principal Component Analysis

Kernel PCA[10] computes the principal components in a high-dimensional feature space, which is nonlinearly related to the input space. The basic idea of kernel PCA is that first map the input data x into a high-dimensional feature space F via a nonlinear mapping Φ and then perform a linear PCA in F. We assume that we are dealing with centered data,i.e., $\sum_{i=1}^{N} \Phi(x_i) = 0$, where N is the number of input data. Kernel PCA diagonalizes the covariance matrix of the mapped data $\Phi(x_i)$

$$C = \frac{1}{N} \sum_{i=1}^{N} \Phi(x_i) \cdot \Phi(x_i). \tag{2}$$

To do this, one has to solve the eigenvalue equation $\lambda v = Cv$ for eigenvalues $\lambda \geq 0$ and $v \in F \setminus \{0\}$. As $Cv = \frac{1}{N} \sum_{i=1}^{N} (\Phi(x_i) \cdot v)\Phi(x_i)$, all solutions v with $\lambda \neq 0$ lie within the span of $\Phi(x_1), \cdots \Phi(x_N)$. Thus there exists coefficients $\alpha_i (i = 1, \ldots, N)$ such that

$$v = \sum_{i=1}^{N} \alpha_i \Phi(x_i) \tag{3}$$

If we consider the following set of equations,

$$\lambda(\Phi(x_i) \cdot v) = (\Phi(x_i) \cdot Cv) \text{ for all } i = 1, \ldots, N. \tag{4}$$

we can substitute (2) and (3) into (4). By defining an $N \times N$ matrix K by $K_{ij} \equiv (\Phi(x_i) \cdot \Phi(x_j))$, we arrive at a problem which is cast in terms of dot products: solve

$$N\lambda\alpha = K\alpha \tag{5}$$

where $\alpha = (\alpha_1, \ldots, \alpha_N)^T$. Normalizing the solutions v^k, i.e.$(v^k \cdot v^k) = 1$ translates into $\lambda_k(\alpha^k \cdot \alpha^k) = 1$. To extract nonlinear principal components of a test data x, we compute the projection onto the k-th component by $\beta_k := (v^k \cdot \Phi(x)) = \sum_{i=1}^{N} \alpha_i^k k(x, x_i)$.

3 Pose Transformation

Our goal is to generate a frontal pose image of unseen test facial image, given its image at certain pose. At first, we extracted features from input face image at each pose using PCA and kernel PCA. Then, we use extracted features to transform an input pose image into its corresponding frontal pose image, i.e. we compute the relation between features at the input pose image and the features at the corresponding frontal pose. A given facial image at each pose can be represented as linear combination of basis vectors at that pose as shown in Fig.2. If we could have the relationship between the coefficient of input pose and that of frontal pose, we can also get the frontal image of input image, because we know each basis vectors of input pose and frontal pose from the training phase. This is our main idea of pose transformation between a given pose and a frontal pose.

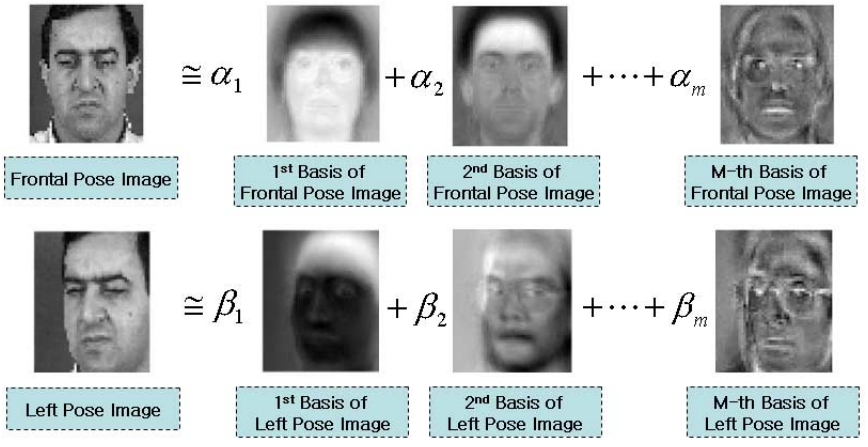


Fig. 2. Representing Input Image by Linear Combination of Basis Vectors

3.1 Obtaining Transformation Matrix by Least Square Estimation

We use m basis functions, Φ^F and Φ^P for subspace representation for each pose image, with $m \leq N$, the number of training images at each pose. From the training set covariance matrix, the Φ^F and Φ^P are known and for a given image at pose P and corresponding frontal image are represented as Fig.2. If we define linear pose transformation matrix between pose P and pose F, $\boldsymbol{U} = [\boldsymbol{u}_1 \cdots \boldsymbol{u}_m]$, we can derive following equation :

$$\alpha_i = \boldsymbol{u}_i^T \boldsymbol{\beta} \tag{6}$$

$$= \boldsymbol{\beta}^T \boldsymbol{u_i} \tag{7}$$

where α_i is i-th coefficient of frontal image, $\boldsymbol{u}_i = (u_{i,1} \cdots u_{i,m})$ is i-th vector of transformation matrix $\boldsymbol{U}$, and $\boldsymbol{\beta} = (\beta_1 \cdots \beta_m)$ is coefficient vector of pose image. We have N training pair images, so we can use least square estimation to solve equation 7. Equation 7 is equivalent to the following equation.

$$\begin{pmatrix} \alpha_i^1 \\ \vdots \\ \alpha_i^N \end{pmatrix} = \begin{pmatrix} \beta_1^1 & \cdots & \beta_m^1 \\ \vdots & \ddots & \vdots \\ \beta_1^N & \cdots & \beta_m^N \end{pmatrix} \begin{pmatrix} u_{i,1} \\ \vdots \\ u_{i,m} \end{pmatrix} \tag{8}$$

where α_i^N is i-th coefficient of N-th frontal image, and β_m^N is m-th coefficient of N-th pose image. This can be written as :

$$A_F = A_L \boldsymbol{u} \tag{9}$$

where

$$A_F = \begin{pmatrix} \alpha_i^1 \\ \vdots \\ \alpha_i^N \end{pmatrix} \tag{10}$$

$$A_L = \begin{pmatrix} \beta_1^1 & \cdots & \beta_m^1 \\ \vdots & \ddots & \vdots \\ \beta_1^N & \cdots & \beta_m^N \end{pmatrix} \tag{11}$$

$$\boldsymbol{u} = \begin{pmatrix} u_{i,1} \\ \vdots \\ u_{i,m} \end{pmatrix} \tag{12}$$

If $A_L^T A_L$ is nonsingular, $\boldsymbol{u}$ is unique and given by

$$\boldsymbol{u} = (A_L^T A_L)^{-1} A_L^T A_F \tag{13}$$

We have m coefficients, accordingly we need to estimate m vectors of $\boldsymbol{u}$ to complete pose transformation matrix $\boldsymbol{U}$. We can obtain pose transformation matrix between input pose and frontal pose using the method we stated. For example, we can write the image of the person at pose L as $I^L = \sum_{i=1}^{m} \alpha_i^L \Phi_i^L$, where Φ_i^L is i-th basis vector for subspace of pose L, and α_i^L is its corresponding coefficient for representing image I^L. Since we have pose transformation matrix between left pose and frontal pose $\boldsymbol{U}_{LF}$, we can represent pose transformed version of input image as follow : $I^F = \sum_{i=1}^{m} \boldsymbol{u}_i^T \boldsymbol{\alpha}^L \Phi_i^F$.

4 Recognition Methods

After pose transformation, we have images transformed to frontal pose. As a result, we can use algorithms generally used for face recognition such as LDA[5], GDA[6], NDA[7][8], and nearest neighbor for experiments. In this section, we review recognition methods we used for experiments.

4.1 LDA and GDA

The face recognition method using LDA is called the fisherface method. It can be applied directly to the gray image[11] [12], or feature vectors of the gray image extracted on a sparse grid[13]. In both cases the classification performance is significantly better than the classification performance of using PCA instead of LDA. LDA is a well-known classical statistical technique using the projection which maximizes the ratio of scatter among the data of different classes to the scatter within the data of the same class[14]. Features obtained by LDA are

useful for pattern classification since they make the data of the same class closer to each other, and the data of different classes further away from each other. Typically, LDA is compared to PCA because both methods are multivariate statistical techniques for projection. PCA attempts to locate the projection that reduces the dimensionality of a data set while retaining as variation in the data set as much as possible[15]. Since PCA does not use class information of data set, LDA usually outperforms PCA for pattern classification.

GDA is nonlinear version of LDA. It generalize LDA to nonlinear problems by mapping the input space into a high dimensional feature space with linear properties. The main idea is to map the input space into a convenient feature space in which variables are nonlinearly related to the input space[6].

4.2 NDA

NDA has similar properties with LDA. It seeks the subspace which can effectively discriminate data classes using within and between class scatter matrices. The main difference is that NDA does not make any assumption about the distribution of data, while LDA assumes each class has gaussian distribution. We briefly review the construction of scatter matrices for NDA, detailed explanation of NDA could be found in [7][8].

For computing nonparametric between scatter matrix, we should obtain extra-class nearest neighbor and intra-class nearest neighbor for all sample points. The extra-class nearest neighbor for a sample x of class C_k is defined as $x^E = \{x^{'} \in \overline{C_k} / \|x^{'} - x\| \leq \|z - x\|, \forall z \in \overline{C_k}\}$ and intra-class nearest neighbor is defined as $x^I = \{x^{'} \in C_k / \|x^{'} - x\| \leq \|z - x\|, \forall z \in C_k\}$. From these neighbors, the extra-class difference and intra-class difference is defined as $\triangle^E = x - x^E$ and $\triangle^I = x - x^I$, respectively. Then the nonparametric between scatter matrix is defined as

$$S^E = \frac{1}{N} \sum_{n=1}^{N} w_n (\triangle_n^E)(\triangle_n^E)^T \tag{14}$$

where N is number of all class samples, $\triangle_n^E$ is the extra-class difference for sample x_n, and w_n is a sample weight defined as

$$w_n = \frac{min\{\| \triangle_n^E \|^\alpha, \| \triangle_n^I \|^\alpha\}}{\| \triangle_n^E \|^\alpha + \| \triangle_n^I \|^\alpha} \tag{15}$$

where α is a control parameter between zero and infinity. The sample weight controls the influence of samples away from class boundaries.

The within-class scatter matrix is defined the same as LDA.

$$S^I = \frac{1}{K} \sum_{k=1}^{K} \Sigma_k \tag{16}$$

Fig. 3. Examples of Database Image

where Σ_k is the class-conditional covariance matrix, estimated from the sample set and K is total number of classes.

5 Experimental Results

5.1 Face Database

We used XM2VT database which consists of 2950 images, i.e. 2 session images of 5 poses for 295 persons. The size of image is 46×56 and each image is aligned by the eye location. The alignment is performed manually. The database contains 5 pose images for each person as shown in Fig.3. We describe each pose as 'F', 'L', 'R', 'U', 'D' respectively.

5.2 Result of Pose Transformation

Fig.4 shows reconstructed pose transformed images of a person. As you can see in Fig.4-(b), transformed images from each pose are similar to each other and they are also similar to the original frontal image. Table.1 represents average reconstruction error of all test images in root-mean squared sense. As expected, the reconstruction error of frontal pose image is the least among 5 poses. But

Fig. 4. (a) Original Images (b) Images Transformed to Frontal Pose

Table 1. Reconstruction Error

	FF	RF	LF	UF	DF
Reconstruction Error	22.5672	30.1874	29.9827	29.6667	30.1382

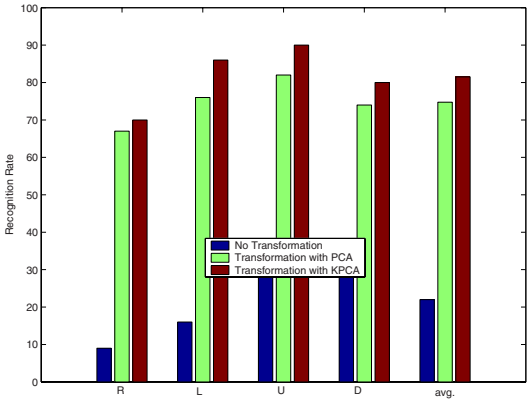


Fig. 5. Recognition Rate with Nearest Neighbor Method

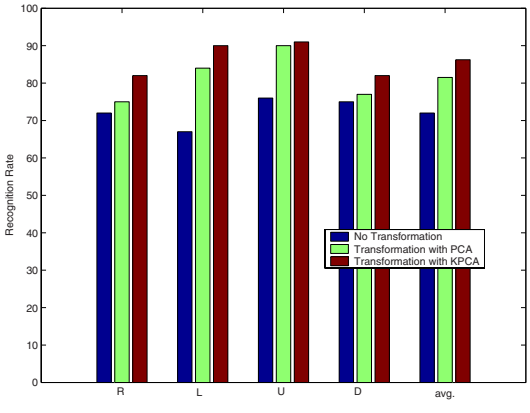


Fig. 6. Recognition Rate with LDA

the reconstruction errors of other pose images are also small enough for each pose image to be used for recognition purpose.

5.3 Recognition Results with Pose Transformed Image

We used 2450 images of 245 persons for training basis vectors of each pose and pose transformation matrix. In addition, we used 500 images of 50 persons for evaluating the recognition performance of pose transformed images. We used

Table 2. Recognition Results

Pose Transformation?	Recognition Method	RF	LF	UF	DF	Avg.
No Trransformation	NN (in input space)	9	16	29	34	22
	NN (in PCA space with 50 dim.)	8	20	23	37	22
	LDA (in PCA space)	72	67	76	75	72
Pose Transformation with PCA Representation	NN (in PCA space with 50 dim.)	67	76	82	74	74.75
	LDA (in PCA space)	75	84	90	77	81.5
	GDA (in PCA space)	67	82	88	79	79
	NDA (in PCA space)	72	76	87	79	78.5
Pose Transformation with KPCA Representation	NN (in kernel space with 200 dim.)	70	86	90	80	81.5
	LDA (in kernel space)	82	90	91	82	86.25
	GDA (in kernel space)	77	85	89	83	83.5
	NDA (in kernel space)	85	89	91	91	89

PCA and Kernel PCA for subspace representation and LDA, GDA, NDA, and nearest neighbor method for face recognition. For recognition experiments, we used two frontal image as gallery and two posed image as probe. For example, RF means that two right posed images are used as probe, two frontal images as gallery.

Table.2 shows the recognition rate. We divided the experiment for 3 parts. First, we checked the recognition rate when no pose transformation is performed. Next, we checked the recognition rate when PCA is used for subspace representation. Finally, we performed the experiment with KPCA as subspace representation. From Fig.5 and Fig.6, you can see recognition rate with pose transformation is much better than no pose transformation case. In addition, we can compare which subspace representation method is better for pose transformation. As you can see in Table.2, recognition rate using kernel PCA is better than PCA case. It is mainly because the nonlinear mapping of KPCA. This makes KPCA represent the input data more effectively than linear PCA. When we use NDA as recognition method, we got the best recognition rate for KPCA case. Because NDA is a nonparametric method, it does not make any assumption about the distribution of data, while LDA assumes each class has gaussian distribution. Moreover, it is uncertain that the transformed frontal pose images of a class are grouped together. Consequently, NDA is a suitable method for this problem. When we used KPCA as subspace representation and NDA as recognition method, we got the best recognition rate.

6 Conclusion

In this paper, we proposed linear pose transformation method in feature space. Our method has some merits, since we used 2D appearance based approach instead of using 3D model based method which requires many preprocessing steps, complicated computing steps, and much execution time. We performed various experiments to show the usefulness of proposed pose transformation method for

face recognition. We compared recognition rate with pose transformation and without pose transformation. Moreover, we compared which subspace representation method is better for pose transformation. According to the experimental results, when we use KPCA as subspace representation and NDA as recognition method, we got the best recognition rate.

References

[1] D.Zhang, D.: AUTOMATED BIOMETRICS-Technologies and Systems. kluwer academic publishers (2000) 211
[2] Beymer, D.: Face recognition under varying pose. In: In Proc. IEEE Conference on Computer Vision and Pattern Recognition. (1994) 756–761 211
[3] Biuk, Z., Loncaric, S.: Face recognition from multi-pose image sequence. In: In Proceedings of 2nd Int'l Symposium on Image and Signal Processing and Analysis. (2001) 211
[4] Huang, F., Zhou, Z., Zhang, H., Chen, T.: Pose invariant face recognition. In: Proceedings of the 4th 1EEE International Conference on Automatic Face and Gesture Recognition. (2000) 245–250 211
[5] Eremad, K., Chellappa, R.: Discriminant analysis for recognition of human face images. Journal of Optical Society of America **14** (1997) 1724–1733 212, 215
[6] Baudat, G., Anouar, F.: Generalized discriminant analysis using a kernel approach. Neural Computation **22** (2000) 2385–2404 212, 215, 216
[7] Bressan, M., Vitria, J.: Nonparametric discriminant analysis and nearest neighbor classification. Pattern Recognition Letters **24** (2003) 2743–2749 212, 215, 216
[8] Fukunaga, K.: Introduction to Statistical Pattern Recognition. Second edn. Academic Press, Boston, MA (1990) 212, 215, 216
[9] Hotelling, H.: Analysis of a compltex statistical variables into principal components. Journal of Educational Psychology **24** (1933) 417–441 212
[10] Scholkopf, B., Smola, A., Muller, K.: Nonlinear component analysis as a kernel eigenvalue problem. Neural Computation **10** (1998) 1299–1319 213
[11] Belhumeur, P., Hespanha, J., Kriegman, D.: 1997. eigenfaces vs. fisherfaces: Class specific linear projection. PAMI **19** (1997) 711–720 215
[12] Etemad, K., Chellappa, R.: Discriminant analysis for recognition of human face images. Journal of Optical Society of America A **14** (1997) 1724–1733 215
[13] Duc, B., Fischer, S., Bigun, J.: Face authentication with gabor information on deformable graphs. IEEE Transactions on Image Processing **8** (1999) 504–516 215
[14] Duda, R., Hart, P., Stork, D.: Pattern Classification. Wiley, New York (2001) 215
[15] Jolliffe, I.: Principal Component Analysis. Springer-Verlag, New York (1986) 216

Model-Based Head and Facial Motion Tracking

F. Dornaika[1] and J. Ahlberg[2]

[1] CNRS HEUDIASYC – UTC, 60205 Compiègne Cedex, France
dornaika@hds.utc.fr
[2] Swedish Defence Research Agency, SE-581 11 Linköping, Sweden
jorahl@foi.se

Abstract. This paper addresses the real-time tracking of head and facial motion in monocular image sequences using 3D deformable models. It introduces two methods. The first method only tracks the 3D head pose using a cascade of two stages: the first stage utilizes a robust feature-based pose estimator associated with two consecutive frames, the second stage relies on a *Maximum a Posteriori* inference scheme exploiting the temporal coherence in both the 3D head motions and facial textures. The facial texture is updated dynamically in order to obtain a simple on-line appearance model. The implementation of this method is kept simple and straightforward. In addition to the 3D head pose tracking, the second method tracks some facial animations using an Active Appearance Model search. Tracking experiments and performance evaluation demonstrate the robustness and usefulness of the developed methods that retain the advantages of both feature-based and appearance-based methods.

1 Introduction

3D head tracking in a video sequence has been recognized as an essential prerequisite for robust facial expression/emotion analysis, face recognition, and model-based image coding. 3D head pose information is also a very important primitive for smart environments and perceptual user interfaces where the poses and movements of body parts need to be determined. With the wide availability of inexpensive cameras and increasingly better support of streaming video by computers, vision-based head and facial motion tracking techniques are well justified. The issue of face recognition and facial analysis has been extensively addressed in recent years [17]. Different approaches including eigenfaces, elastic graph models, deformable templates [18], Active Shape Models [6] and Active Appearance Models [5] have shown to be promising under different assumptions. A huge research effort has been devoted to detecting and tracking of head and facial features in 2D and 3D (e.g., [15, 12]). Most tracking approaches take advantage of the constrained scenario: the face and/or the head are not viewed as arbitrary tracked objects. A model-based approach is favored [10, 13, 16]. When there is no expression change on the face, relative head pose can be solved as a rigid object tracking problem through traditional 3D vision algorithms for multiple view analysis [11]. However, in practice, expressional deformation or even occlusion

N. Sebe et al. (Eds.): HCI/ECCV 2004, LNCS 3058, pp. 221–232, 2004.

frequently occurs, together with head pose changes. Therefore, it is necessary to develop effective techniques for head tracking under these conditions.

In this paper, we propose two methods. The first method only tracks the 3D head pose by combining three concepts: (i) a robust feature-based pose estimator by matching two consecutive frames, (ii) a featureless criterion utilizing an on-line appearance model for the facial texture (temporal coherence of facial texture), and (iii) a temporal coherence of 3D head motions. The first two criteria do not need any prior training. The third criterion, however, needs some prior knowledge on the dynamics of head motions. For example, this prior can be built from experiments reporting the dynamics of head motions (see [14]).

In addition to the 3D head pose tracking, the second method tracks some facial features using an Active Appearance Model search. The rest of the paper is organized as follows. Section 2 introduces the deformable model. Section 3 describes the proposed real-time head tracking scheme. Section 4 describes the tracking of facial features using active appearance model search. Section 5 presents some experimental results.

2 A Deformable Model

2.1 A Parameterized 3D Face Model

Building a generic 3D face model is a challenging task. Indeed, such a model should account for the differences between different specific human faces as well as between different facial expressions. This modelling was explored in the computer graphics, computer vision, and model-based image coding communities (e.g., see [3]). In our study, we use the 3D face model *Candide*. This 3D deformable wireframe model was first developed for the purpose of model-based image coding and computer animation. The 3D shape is directly recorded in coordinate form. The shape is given by a set of vertices and triangles. The 3D face model is given by the 3D coordinates of the vertices $\mathbf{P}_i, i = 1, \cdots, n$ where n is the number of vertices. Thus, the shape up to a global scale can be fully described by the $3n$-vector $\mathbf{g}$ – the concatenation of the 3D coordinates of all vertices $\mathbf{P}_i$. The vector $\mathbf{g}$ can be written as:

$$\mathbf{g} = \overline{\mathbf{g}} + \mathbf{S}\,\sigma + \mathbf{A}\,\alpha \tag{1}$$

where $\overline{\mathbf{g}}$ is the standard shape of the model, and the columns of $\mathbf{S}$ and $\mathbf{A}$ are the Shape and Animation Units, respectively. The shape and animation variabilities can be approximated well enough for practical purposes by this linear relation. A Shape Unit provides a way to deform the 3D wireframe such as to make the eye width bigger, head wider, etc. Without loss of generality, we have chosen the following Action Units [8]: 1) Jaw drop, 2) Lip stretcher, 3) Lip corner depressor, 4) Upper lip raiser, 5) Eyebrow lowerer, 6) Outer eyebrow raiser. These Action Units are enough to cover most common facial expressions (mouth and eyebrow movements). More details about this face model can be found in [7].

Since σ is person dependent it can be computed once using either feature-based or appearance-based approaches. Therefore, the geometry of the 3D wireframe can be written as ($\mathbf{g}_s = \overline{\mathbf{g}} + \mathbf{S}\,\sigma$ is the static model):

$$\mathbf{g} = \mathbf{g}_s + \mathbf{A}\,\alpha \tag{2}$$

2.2 Projection Model and 3D Pose

The adopted projection model is the weak perspective projection model [2]. Therefore, the mapping between the 3D face model and the image is given by a 2×4 matrix $\mathbf{M}$. Thus a 3D vertex $\mathbf{P}_i = (X_i, Y_i, Z_i)^T \subset \mathbf{g}$ will be projected onto the image point $\mathbf{p}_i = (u_i, v_i)^T$ given by:

$$(u_i, v_i)^T = \mathbf{M}\,(X_i, Y_i, Z_i, 1)^T \tag{3}$$

Let $\mathbf{R} \equiv \{r_{lj}\}\, l, j = 1, 2, 3$ and $\mathbf{t} = (t_x, t_y, t_z)^T$ be the rotation and translation between the 3D face model coordinate system and the camera coordinate system. Let α_u, α_v, u_c, and v_c be the intrinsic parameters of the camera. The factor α_u (α_v) is the focal length of the camera expressed in horizontal (vertical) pixels. u_c and v_c are the coordinates of the principal point (image center). The 2×4 projection matrix $\mathbf{M}$ is given by:

$$\mathbf{M} = \begin{pmatrix} \frac{\alpha_u}{t_z}\, s\, \mathbf{r}_1^T & \frac{\alpha_u}{t_z}\, t_x + u_c \\ \frac{\alpha_v}{t_z}\, s\, \mathbf{r}_2^T & \frac{\alpha_v}{t_z}\, t_y + v_c \end{pmatrix}$$

where $\mathbf{r}_1^T$ and $\mathbf{r}_2^T$ are the first two rows of the rotation matrix $\mathbf{R}$, and s is an unknown scale (the *Candide* model is given up to a scale). Without loss of generality, we can assume that the aspect ratio is equal to one, yielding:

$$\mathbf{M} = \begin{pmatrix} s_u\, \mathbf{r}_1^T & \lambda_u\, t_x + u_c \\ s_u\, \mathbf{r}_2^T & \lambda_u\, t_y + v_c \end{pmatrix} \tag{4}$$

We can thus easily retrieve the pose parameters from the projection matrix, and vice versa. We represent the rotation matrix $\mathbf{R}$ by the three Euler angles θ_x, θ_y, and θ_z. In the sequel, the 3D pose (global motion) parameters can be represented by the 6-vector $\mathbf{z}_g = [\theta_x, \theta_y, \theta_z, s_u, t'_x, t'_y]^T$ where $t'_x = \lambda_u\, t_x$ and $t'_y = \lambda_u\, t_y$. Note that the face is allowed to move along the depth direction since t_z is implicitly accounted for in $s_u = \frac{\alpha_u}{t_z}\, s$. Thus, the geometry of the model is parameterized by the parameter vector $\mathbf{b}$:

$$\mathbf{b} = [\mathbf{z}_g^T, \sigma^T, \alpha^T]^T = [\theta_x, \theta_y, \theta_z, s_u, t'_x, t'_y, \sigma^T, \alpha^T]^T$$

For a given person, only $\mathbf{z}_g$ and α are time dependent.

2.3 Geometrically Normalized Facial Images

A face texture is represented as a geometrically normalized image, i.e. a *shape-free texture*. The geometry of this image is obtained by projecting the standard

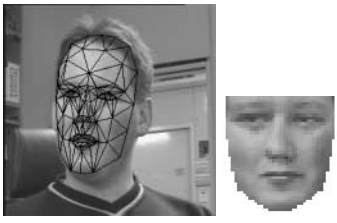


Fig. 1. An input images with correct adaptation (left). The corresponding geometrically normalized image (right)

shape $\overline{\mathbf{g}}$ (wireframe) using a standard 3D pose (frontal view) onto an image with a given resolution (intrinsic parameters). This geometry is represented by a triangular 2D mesh. The texture of this geometrically normalized image is obtained by texture mapping from the triangular 2D mesh in the input image using a piece-wise affine transform. For a very fast texture mapping (image warping), we have exploited the fact that the 2D geometry of the destination mesh can be known in advance.

In fact, the geometrical normalization normalizes three different things: the 3D head pose, the facial animation, , and the geometrical differences between individuals. Mathematically, the warping process applied to an input image $\mathbf{y}$ is denoted by:

$$\mathbf{x}(\mathbf{b}) = \mathcal{W}(\mathbf{y}, \mathbf{b}) \tag{5}$$

where $\mathbf{x}$ denotes the geometrically normalized texture and $\mathbf{b}$ denotes the geometrical parameters. $\mathcal{W}$ is the piece-wise affine transform. Figure 1 displays the geometrical normalization result associated with an input image (256×256) having a correct adaptation. The geometrically normalized image is of resolution 40×42.

We point out that for close ranges of the head with respect to the camera, the 3D tracking based on the *shape-free texture* obtained with the *Candide* model is expected to be more accurate than the tracking based on the texture maps associated with cylindrical face models [4].

3 Head Tracking

Given an image of a face (or a video sequence), the head tracking consists in estimating the 3D head pose, i.e. the vector $\mathbf{z}_g$ (or equivalently the projection matrix $\mathbf{M}$), for each image. In our work, this estimation is carried out independently of the animation parameters encoded by the vector α.

The outline of the method allowing the recovery of the 3D head pose is illustrated in Figure 2. The method consists of a cascade of two stages. The first stage uses a RANSAC-based method [9] to infer the pose by matching features in two consecutive frames. This is responsible for the accuracy of the 3D head pose. Also, this stage provides a tractable set of plausible solutions that will be

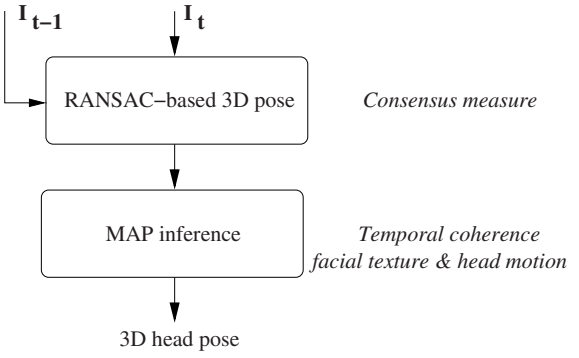


Fig. 2. 3D head pose recovery for the frame t. The first stage exploits the rigidity of head motions to retrieve a set of candidate solutions by matching features in the two consecutive frames. The second stage exploits the temporal coherence of the facial texture and the 3D head motion

processed by the second stage. The second stage (two criteria used by a MAP inference) exploits the temporal coherence of both the 3D head motions and the facial textures. This stage diminishes the effects of gradual lighting changes since the *shape-free texture* is dynamically updated. Also, this stage criterion prevents the tracker from drifting (error accumulation).

In the rest of this section, we give the details of the proposed method.

Given two consecutive images, I_{t-1} and I_t, of a face undergoing rigid and non-rigid motion (head motion and facial animation), one can still find a set of facial features which are only affected by the rigid motion. Features undergoing local deformation/animation can be considered as outliers. Thus, by identifying the inlier features (whose motion is fully described by the rigid motion), the projection matrix/3D pose can be recovered.

Computing the projection matrix using the random sampling technique RANSAC requires a set of correspondences between the 3D face model and the current image I_t. Since a direct match between 3D features and 2D images is extremely difficult, we exploit the adaptation (3D pose and 3D shape) associated the old frame I_{t-1} and project the 3D vertices of the model onto it. Notice that this adaptation is already known.

In the experiments shown below, we have kept 101 vertices belonging to the central part of the 3D model. The patches of I_{t-1} centered on the obtained projections are then matched with the current image using the Zero Mean Normalized Cross Correlation with sub-pixel accuracy within a certain search region. The computed matches $\mathbf{p}_{i(t-1)} \leftrightarrow \mathbf{p}_{i(t)}$ will give the set of 3D-to-2D correspondences $\mathbf{P}_{i(t-1)} \leftrightarrow \mathbf{p}_{i(t)}$ which will be handed over to the RANSAC technique.

The 2D matching process is made reliable and fast by adopting a multi-stage scheme. First, three features are matched in the images I_{t-1} and I_t (the two inner eye corners and the philtrum top) from which a 2D affine transform is computed

between I_{t-1} and I_t. Second, the 2D features $\mathbf{p}_{i(t-1)}$ are then matched in I_t using a small search window centered on their 2D affine transform.

3.1 The Algorithm

Retrieving the projection matrix $\mathbf{M}$ from the obtained set of putative 3D-to-2D correspondences is carried out using two stages (see Figure 2). The first stage (*Exploration stage*) explores the set of 3D-to-2D correspondences using the conventional RANSAC paradigm [9]. The second stages (*Search stage*) selects the solution by integrating the consensus measure and a MAP inference based on the temporal coherence of both the facial texture and the head motion. As a result the computation of the 3D head pose was guided by three independent criteria. The goals of these criteria are: (i) removing possible mismatches and locally deformed features from the computation of the projection matrix, (ii) obtaining an accurate 3D head pose, and (iii) preventing the tracker from drifting.

Once the putative set of 3D-to-2D correspondences is known, the proposed method can be summarized as follows. Let n_c be the total number of correspondences. For the sake of simplicity the subscript (t) has been omitted.

First stage: consensus measure

1. Randomly sample four 3D-to-2D feature correspondences $\mathbf{P} \leftrightarrow \mathbf{p}$ (non-coplanar configuration). The image points of this sample are chosen such that the mutual distance is large enough.
2. Compute the matrix $\mathbf{M}$ using this sample.
3. For all feature correspondences, compute the distance between the image features $\mathbf{p}$ and the projection $\mathbf{M}\,\mathbf{P}$.
4. Count the number of features for which the distance is below some threshold. This consensus measure is denoted by n_I. A threshold value between 1.0 and 2.0 pixels works well for our system.

In our work, the number of random samples is capped at the number of feature correspondences n_c. In our experiments, n_c is variable since matches are filtered out by thresholding their normalized cross-correlation. Typically, n_c is between 70 and 101 features assuming we have used 101 features.

Second stage: MAP inference

1. Sort the projection matrices according to their n_I in a descending order.
2. For the best hypotheses (e.g., 10 solutions), refit the matrix $\mathbf{M}$ using its inliers.
3. For each such hypothesis, compute the associated unnormalized a-posterior probability as (m denotes the measurement):

$$p(\mathbf{M}_i|m) \propto p(m|\mathbf{M}_i)p(\mathbf{M}_i)$$

The term $p(\mathbf{M}_i)$ represents the prior associated with the hypothesis $\mathbf{M}_i$ (3D head pose). Since the 3D head pose is tracked, this prior can describe

the temporal change of the head pose, that is, this prior can set to the $p(\mathbf{M}_{i(t)}|\mathbf{M}_{(t-1)})$. If one assumes that the tracking rate is high, i.e. the head motions are small between frames, then one plausible expression of this prior can be [14]:

$$p(\mathbf{M}_i) \propto \exp\left(-\frac{||\mathbf{v}_i||^2}{2\sigma_v^2} - \frac{||\mathbf{w}_i||^2}{2\sigma_w^2}\right)$$

where $\mathbf{v}_i$ and $\mathbf{w}_i$ are the translational and rotational velocities of the head implied by the pose hypothesis $\mathbf{M}_i$ and σ_v and σ_w are the learned standard deviations of these quantities.

The second term is a likelihood measurement and should tell how well the associated image measurement is consistent with the current hypothesis $\mathbf{M}_i$. As a measurement, we choose the normalized correlation, ρ_i, between the associated texture, $\mathbf{x}(\mathbf{M}_i)$, and the current appearance model of the facial texture $\mathbf{A}_t$ summarizing the facial appearances up to time $t-1$. Thus, we can write:

$$p(m|\mathbf{M}_i) = \frac{(1+\rho_i)/2}{\sum_i (1+\rho_i)/2}$$

Note that the measure $p(m|\mathbf{M}_i)$ also quantifies the consistency of the current texture with the face texture.

4. Select the $\mathbf{M}$ (3D head pose) which has the maximum a-posterior probability, i.e.:

$$\mathbf{M}^\star = \arg\max_{\mathbf{M}_i}(p(\mathbf{M}_i|m))$$

5. The appearance $\mathbf{A}_t$ can be updated as

$$\mathbf{A}_t = (1-\lambda)\,\mathbf{A}_{t-1} + \lambda\,\mathbf{x}(\mathbf{M}^*_{t-1}) \tag{6}$$

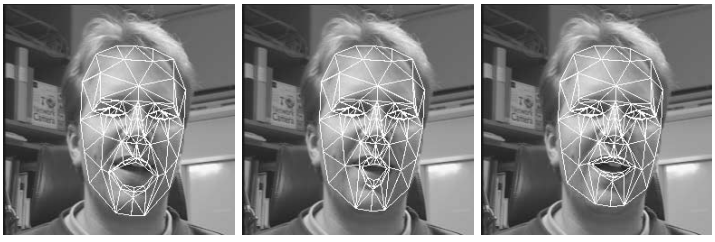
With this updating scheme, the old information stored in the model decays exponentially over time.

4 Head and Facial Motion Tracking

When the facial motion should be determined in addition to the 3D head pose, we proceed as follows. Once the 3D head pose is recovered for a given input frame, we aim at estimating the associated animation parameters, i.e. the vector α. To this end, we use the concept of the active appearance model search [1, 5]. This concept aims at finding the geometric parameters by minimizing the residual error between a learned face appearance and the synthesized appearance. In our case, the geometric parameters are only the components of the vector α.

5 Experimental Results

Before detailed discussion of tracking experiments, we present an example showing how the proposed methods work. Figure 3 displays the adaptation (head and



(a) **(b)** **(c)**

Fig. 3. The adaptation process applied to an input image at different stages. **(a)** 3D head pose computation. **(b)** and **(c)** facial motion computation (first iteration and convergence of the AAM search)

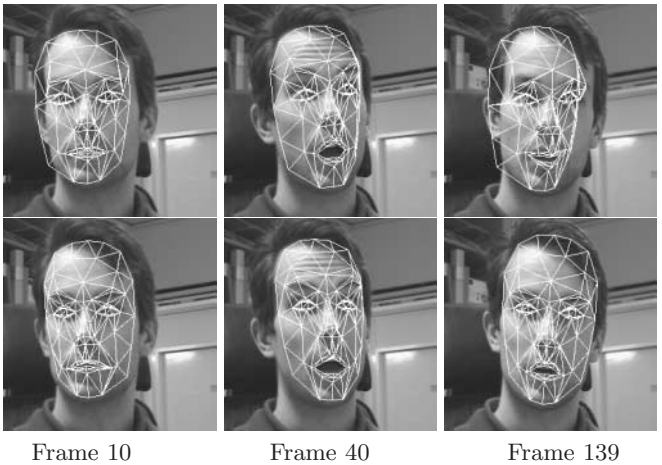
Fig. 4. Tracking the head motion using a test sequence of 340 frames. Facial animation is not computed. The right-bottom of the figure displays the 340^{th} frame

facial animation) to an input image at several stages of the proposed method. **(a)** displays the computed 3D head pose using the RANSAC technique and the MAP inference (Section 3). **(b)** displays the facial motion estimation (animation parameters) obtained at the third iteration of the AAM search algorithm. **(c)** displays the results obtained at the sixth iteration (convergence). Note that the 3D model has been correctly adapted to the head motion using the 3D pose computation stage (Section 3) while the mouth animation (a local motion) has been correctly computed by the iterative search algorithm.

5.1 Tracking Experiments and Accuracy Evaluation

Figure 4 shows the tracking results based on a test sequence of 340 frames using the framework described in Section 3 (head motion tracking). One can notice that the proposed tracker has succeeded to accurately track the 3D head pose despite the presence of large facial expressions (local motion).

Fig. 5. Tracking the 3D head pose and the facial animation using a test sequence of 340 frames. The right-bottom of the figure displays the 340^{th} frame



Frame 10 Frame 40 Frame 139

Fig. 6. Top: applying the RANSAC technique alone to a video of 140 frames. Bottom: applying a RANSAC technique with a MAP inference to the same video sequence (our proposed method)

Figure 5 shows the tracking results based on the same test sequence using the proposed framework described in Sections 3 and 4 (3D head pose and facial animation tracking). In this case, not only the 3D head pose is computed but also the facial animation associated with six FACS (the vector α) is computed using an AAM search.

Figure 6 shows the tracking results based on another test sequence of 140 frames. The top of this figure displays the tracking results when the 3D head pose is computed with a conventional feature-based RANSAC technique. Note

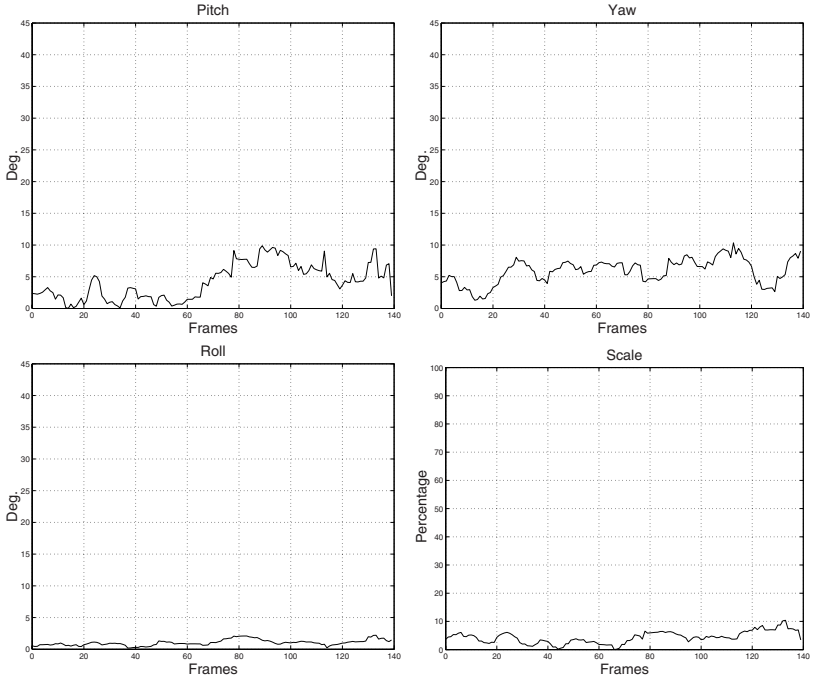


Fig. 7. 3D pose errors using a synthesized sequence (pitch, yaw, roll, and scale). For each frame in the synthetic sequence and for each parameter, the error is the absolute value of the difference between the estimated value and its ground truth value used in animating the synthetic images

that a conventional RANSAC technique selects the solution which corresponds to the highest number of inlier features. The bottom of this figure displays the results of applying our proposed method (Sections 3 and 4) to the same sequence. For both methods, the adaptation is displayed for frames 10, 40, and 139. As can be seen, the RANSAC-based tracking suffers from some drifting due to the 3D model inaccuracies which has not occurred in our method that combines the RANSAC technique with a MAP inference.

Accuracy Evaluation. Figure 7 displays the 3D pose errors associated with a synthesized sequence of 140 frames. The ground truth associated with this sequence is known. The plots correspond to the three Euler angles and the scale. As can be seen, the developed tracker is accurate enough to be useful in many applications. The non-optimized implementation of the tracking algorithm (3D head pose and facial animation) takes about 30 ms per frame. We have used the C language and the Unix Operating System. The developed tracker can handle out-of-plane rotations (pitch and yaw angles) within the interval $[-40^o, 40^o]$. The main mode of failure is when the tracker is faced with ultra-rapid movements (global or local) such that the feature-based matching and/or the appearance-based facial animation tracking can loose track.

6 Conclusion

We have addressed the real-time tracking of head and facial motion in monocular image sequences. We decouple the 3D head pose estimation from the estimation of facial animation. This goal is attained in two phases and gives rise to two methods (the first method only utilizes the first phase). In the first phase, the 3D head pose was computed by combining a RANSAC-based technique with a MAP inference integrating temporal consistencies about the face texture and 3D head motions. In the second phase, the facial motion was computed using the concept of the active appearance model search.

References

[1] J. Ahlberg. An active model for facial feature tracking. *EURASIP Journal on Applied Signal Processing*, 2002(6):566–571, June 2002. 227

[2] Y. Aloimonos. Perspective approximations. *Image and Vision Computing*, 8(3):177–192, 1990. 223

[3] V. Blanz and T. Vetter. A morphable model for the synthesis of 3D faces. In *Proc. SIGGRAPH'99 Conference*, 1999. 222

[4] M. L. Cascia, S. Sclaroff, and V. Athitsos. Fast, reliable head tracking under varying illumination: An approach based on registration of texture-mapped 3D models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(4):322–336, 2000. 224

[5] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):681–684, 2001. 221, 227

[6] T. F. Cootes, C. J. Taylor, D. Cooper, and J. Graham. Active shape models–their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995. 221

[7] F. Dornaika and J. Ahlberg. Face and facial feature tracking using deformable models. *International Journal of Image and Graphics*, July 2004. 222

[8] P. Ekman and W. V. Friesen. *Facial Action Coding System*. Consulting Psychology Press, Palo Alto, CA, USA, 1977. 222

[9] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communication ACM*, 24(6):381–395, 1981. 224, 226

[10] S. B. Gokturk, J. Y. Bouguet, C. Tomasi, and B. Girod. Model-based face tracking for view-independent facial expression recognition. In *Proc. Face and Gesture Recognition*, 2002. 221

[11] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000. 221

[12] N. Olivier, A. Pentland, and F. Berard. Lafter: Lips and face real time tracker. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 1997. 221

[13] F. Preteux and M. Malciu. Model-based head tracking and 3D pose estimation. In *Proc. SPIE Conference on Mathematical Modeling and Estimation Techniques in Computer Vision*, 1998. 221

[14] D. Reynard, A. Wildenberg, A. Blake, and J. Marchant. Learning the dynamics of complex motions from image sequences. In *Proc. European Conference on Computer Vision*, 1996. 222, 227

[15] J. Ström. Model-based real-time head tracking. *EURASIP Journal on Applied Signal Processing*, 2002(10):1039–1052, 2002. 221

[16] H. Tao and T. Huang. Explanation-based facial motion tracking using a piecewise Bezier volume deformation model. In *Proc. IEEE Computer Vision and Pattern Recognition*, 1999. 221

[17] M. H. Yang, D. J. Kriegman, and N. Ahuja. Detecting faces in images: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1):34–58, 2002. 221

[18] A. L. Yuille, D. S. Cohen, and P. Hallinan. Feature extraction from faces using deformable templates. *International Journal of Computer Vision*, 8(2):99–112, 1992. 221

Author Index